

Improving Bespoke Software Projects Estimation in South Africa

Sebastian Wade

**A research report submitted to the Faculty of Commerce, Law and
Management, University of the Witwatersrand, in partial fulfilment of the
requirements for the degree of Master of Business Administration**

Johannesburg, 2010

ABSTRACT

Software development has had much comment over the last two to three decades; large research organisations such as the Standish Group have carried out several empirical studies that measure the success rate of software projects as they have typically been unsuccessful over time. An integral part of any software development project is the software estimation process which establishes project scope; within the scope of the project are a number of key parameters which assist in mapping out the entire project.

The objective of this research report is to focus on software estimation in South Africa and how software practitioners in the country are approaching the formidable task of software estimation with the aim of improving this process and the number of successful software projects. Bespoke software development is the focus area of software development in the report. The research was conducted by way of one-on-one interviews with software practitioners who had bespoke software development knowledge relative to a certain size project value and experience. The report findings represent the perceptions and viewpoints of the respondents.

It was established from the findings that the approach to software estimation in South Africa is largely done on an informal or unstructured basis, that estimation which is done on a more formal basis is done using the expert or experience based technique. The respondents were only able to identify two other forms of estimation, neither of which appears to be used in South Africa.

The influence of business interests upon producing the final estimate and price to the customer is powerful and often compromises the objective of the estimate. South African software vendors do however employ the best practice of revising their estimates after inception of the project and are able to manage the uncertainty element of the project more effectively in this way. Given the relative inexperience of the software vendor community in South Africa they were somewhat unaware of the international practice of combining estimation

techniques even though they agreed that it may be a solution towards improving the estimation process overall.

South Africa appears to be uneducated as to the majority of theory and application surrounding the use of various estimation techniques. This fact may simply be because the country is still developing behind first world countries that have already overcome this learning curve and progressed from that point. While there are many experienced software practitioners in South Africa there is no indication of the estimation processes evolving so that software estimation can progress to a more professional, formalised level.

DECLARATION

I, Sebastian Wade, declare that this research report is my own work except as indicated in the references and acknowledgements. It is submitted in partial fulfilment of the requirements for the degree of Master of Business Administration in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in this or any other university.

Sebastian Wade

Signed at

On the day of 2010

DEDICATION

This research report is dedicated first and foremost to my wife, Wendy. Secondly, to my two young sons, Cameron and Alexander, who were born during my period of study and have illuminated our lives and, of course, to my Mom.

“Better late than never...”

ACKNOWLEDGEMENTS

Thank you to my wife and children who provided me with the much needed love, support and understanding one needs to complete a task such as this, to my Mom for her gentle encouragement and for believing in me. To the rest of my friends and family that showed their support and provided assistance in one way or another over the last few years while I have been busy with my studies. A special thanks to Althea Bacchialoni for her guidance at a critical time. Many thanks to those work colleagues who filled in while I was away and for being understanding while I was distracted from my responsibilities at the office.

To all those respondents that assisted me with my research and provided me with their valuable time and experience, I hope and trust that I have done you all justice and that hopefully my research in some small way may bring about some positive change. All your input is sincerely appreciated.

Finally to the Wits Business School and all the faculty members that devoted their time in imparting their vast knowledge and expertise to me, I believe that I have grown in so many ways as a result of it and would or could never forget the time and invaluable lessons learnt during this period. A warm thank you to my supervisor Antony Soicher who has seen me through this period, he has providing the necessary guidance and advice when needed and been patient during my absences.

TABLE OF CONTENTS

ABSTRACT	II
DECLARATION.....	IV
DEDICATION	V
ACKNOWLEDGEMENTS.....	VI
LIST OF TABLES.....	X
LIST OF FIGURES	X
CHAPTER 1: INTRODUCTION	1
1.1 PURPOSE OF THE STUDY	1
1.2 CONTEXT OF THE STUDY.....	1
1.3 PROBLEM STATEMENT	3
1.4 SIGNIFICANCE OF THE STUDY	3
1.5 DELIMITATIONS OF THE STUDY.....	3
1.6 DEFINITION OF TERMS	5
1.7 ASSUMPTIONS	5
CHAPTER 2: LITERATURE REVIEW	7
2.1 INTRODUCTION	7
2.2 BACKGROUND DISCUSSION	9
2.3 KEY ELEMENTS OF MANAGING SOFTWARE DEVELOPMENT PROJECTS.....	10
2.3.1 SOFTWARE DEVELOPMENT TRENDS.....	10
2.3.2 UNDERLYING CAUSES OF SOFTWARE DEVELOPMENT FAILURE	11
2.3.3 CRITICAL SUCCESS FACTORS OF SOFTWARE DEVELOPMENT	12
2.3.4 CONCLUSION TO SECTION 2.3.....	14
2.4 PROJECT MANAGEMENT OF SOFTWARE DEVELOPMENT	14
2.4.1 SOFTWARE PROJECT MANAGEMENT (SPM)	15
2.4.2 SPM CHALLENGES AND FOCUS AREAS	16
2.4.3 CONCLUSION TO SECTION 2.4.....	18
2.5 SOFTWARE ESTIMATION	19
2.5.1 THE ROLE OF SOFTWARE ESTIMATION	19
2.5.2 PROBLEMS ENCOUNTERED WITHIN SOFTWARE ESTIMATION	20
2.5.3 TYPES OF SOFTWARE ESTIMATION AND A COMPARISON OF METHODS	23
2.5.4 ESTIMATION BEST PRACTICES	26
2.5.5 CONCLUSION TO SECTION 2.5.....	28
2.6 CONCLUSION OF LITERATURE REVIEW	29

2.6.1	PROPOSITION 1: EXPERT ESTIMATION TECHNIQUES ARE PREFERRED TO MODEL-BASED TECHNIQUES IN SOUTH AFRICA.....	31
2.6.2	PROPOSITION 2: BUSINESS INTERESTS AS OPPOSED TO PROJECT SCOPE ARE THE GREATEST INFLUENCE IN ARRIVING AT SOFTWARE ESTIMATES	31
2.6.3	PROPOSITION 3: ESTIMATE VALUES ARE COMMITTED TO AT THE START OF THE PROJECT AND ARE SELDOM REVISED.....	31
2.6.4	PROPOSITION 4: DESPITE A COMBINATION OF ESTIMATION TECHNIQUES BEING ADVOCATED BY ACADEMICS THIS IS SELDOM PRACTISED IN REALITY.....	31

CHAPTER 3: RESEARCH METHODOLOGY 32

3.1	RESEARCH METHODOLOGY /PARADIGM	32
3.2	RESEARCH DESIGN	33
3.3	POPULATION AND SAMPLE.....	34
3.3.1	POPULATION.....	34
3.3.2	SAMPLE AND SAMPLING METHOD	34
3.4	THE RESEARCH INSTRUMENT	35
3.5	PROCEDURE FOR DATA COLLECTION.....	36
3.6	DATA ANALYSIS AND INTERPRETATION	36
3.7	LIMITATIONS OF THE STUDY.....	37
3.8	VALIDITY AND RELIABILITY	38
3.8.1	EXTERNAL VALIDITY.....	38
3.8.2	INTERNAL VALIDITY	38
3.8.3	RELIABILITY	39

CHAPTER 4: PRESENTATION OF RESULTS 40

4.1	INTRODUCTION	40
4.2	DEMOGRAPHIC PROFILE OF RESPONDENTS	40
4.3	RESULTS PERTAINING TO PROPOSITION 1	42
4.4	RESULTS PERTAINING TO PROPOSITION 2	44
4.5	RESULTS PERTAINING TO PROPOSITION 3.....	49
4.6	RESULTS PERTAINING TO PROPOSITION 4.....	50
4.7	SUMMARY OF THE RESULTS	52
4.7.1	DEMOGRAPHIC PROFILE OF RESPONDENTS	52
4.7.2	PROPOSITION 1 SUMMARY	53
4.7.3	PROPOSITION 2 SUMMARY	53
4.7.4	PROPOSITION 3 SUMMARY	55
4.7.5	PROPOSITION 4 SUMMARY	55

CHAPTER 5: DISCUSSION OF THE RESULTS.....56

5.1	INTRODUCTION	56
5.2	DEMOGRAPHIC PROFILE OF RESPONDENTS	56
5.3	DISCUSSION PERTAINING TO PROPOSITION 1	57
5.3.1	CONCLUSION – PROPOSITION 1.....	59
5.4	DISCUSSION PERTAINING TO PROPOSITION 2.....	60
5.4.1	CONCLUSION – PROPOSITION2.....	61

5.5	DISCUSSION PERTAINING TO PROPOSITION 3.....	62
5.5.1	CONCLUSION – PROPOSITION 3.....	64
5.6	DISCUSSION PERTAINING TO PROPOSITION 4.....	64
5.6.1	CONCLUSION – PROPOSITION 4.....	65
5.7	CONCLUSION	66

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS..... 68

6.1	INTRODUCTION	68
6.2	CONCLUSIONS OF THE STUDY	68
6.2.1	PROPOSITION 1: EXPERT ESTIMATION TECHNIQUES ARE PREFERRED TO MODEL-BASED TECHNIQUES IN SOUTH AFRICA.....	68
6.2.2	PROPOSITION 2: BUSINESS INTERESTS AS OPPOSED TO PROJECT SCOPE ARE THE GREATEST INFLUENCE IN ARRIVING AT SOFTWARE ESTIMATES	69
6.2.3	PROPOSITION 3: ESTIMATE VALUES ARE COMMITTED TO AT THE START OF THE PROJECT AND ARE SELDOM REVISED.....	70
6.2.4	PROPOSITION 4: DESPITE A COMBINATION OF ESTIMATION TECHNIQUES BEING ADVOCATED BY ACADEMICS THIS IS SELDOM PRACTISED IN REALITY.....	70
6.3	RECOMMENDATIONS	71
6.4	SUGGESTIONS FOR FURTHER RESEARCH	73

REFERENCES 74

APPENDIX A..... 79

INTERVIEW QUESTIONNAIRE	79
-------------------------------	----

APPENDIX B..... 89

CONSISTENCY MATRIX.....	89
-------------------------	----

LIST OF TABLES

Table 1: Terms and Meanings used in the Report	5
Table 2: Advantages and Disadvantages of Model and Expertise-Based Techniques.....	25
Table 3: Profile of Respondents	35
Table 4: Respondent Names and Organisations.....	35
Table 5: Profile of Intended Respondents	40
Table 6: Skills Matrix for Respondents Interviewed.....	41
Table 7: Respondent Experience of the Use of Estimation Techniques in South Africa.....	43
Table 8: Motivation for Software Estimates being too Low.....	46
Table 9: Revision of the Software Estimation during the Project's Progress....	49

LIST OF FIGURES

Figure 1: Diagrammatic View of Literature Topics and Sub-Topics.....	8
Figure 2: The Cone of Uncertainty (McConnell 2006)	21
Figure 3: Software Estimation Techniques (Boehm, Abts and Chulani 2000)..	23
Figure 4: Project Experience by Respondent in Rand Value	41
Figure 5: Named Techniques Employed By or Known to Respondents.....	42

Figure 6: Estimation Technique prevalence in the South African Marketplace	44
Figure 7: Level of Customer Involvement in the Estimation Process	45
Figure 8: Role of Estimation Process and Benefit to Provider and Acquirer	45
Figure 9: Clients Financial Needs/Constraints and the Influencing Factor	47
Figure 10: Respondents Familiar with the Combining of Estimation Techniques	50
Figure 11: Respondents that agreed with the statement that the Combination of Estimation Techniques is advocated by Academics but seldom practised in Reality	51
Figure 12: Respondents that believed that the Combination of Estimation Techniques provides more reliable Estimates.....	51
Figure 13: Respondents that believe it is Possible to Combine Estimation Techniques in Reality.....	52
Figure 14: The Cone of Uncertainty (McConnell 2006)	63

CHAPTER 1: INTRODUCTION

1.1 Purpose of the study

The purpose of the research was to gain insight into software estimation and the different estimation approaches employed in South Africa, evaluate them in relation to the existing literature, extract a meaningful analysis and make a positive contribution towards improving the process of software estimation in South Africa. The research focused on bespoke software development and implementation projects that arose between the software provider and the customer that required a specific software solution to meet a specific business need or needs.

1.2 Context of the study

“Software is one of the most important and yet one of the most economically challenging technologies in the current era. As a purely intellectual product, it is among the most labour intensive, complex, and error-prone technologies in human history.” (Mahanti and Antony 2005: 739)

The Standish Group released “The CHAOS Report” in 1994 and identified software development as being predominantly over time and budget, their research found that 52.7% of the projects cost 189% of their original estimates as opposed to only 16.2% of all projects that were completed on time and budget. Furthermore it stated that the software itself was often unreliable and that a primary reason for this was that no one was making an attempt to learn from their mistakes (Standish Group 1994). Considering the software industry produced 600 billion dollars worth of software worldwide in 2002, (McManus 2004) the problem needs to be addressed in an attempt to curtail the massive amount of wasted time, money and effort that is being spent worldwide.

The opening quotation by Mahanti and Antony suggests that the key to successful software development had not yet been fully answered by 2005, 11 years subsequent to the original CHAOS report. This is not to say that software

development and Information Technology (IT) projects in general have not received sufficient attention and subsequently improved. There are many theories and models that have already been developed in an attempt to minimise the failure rate of these projects. A project planning model intended to manage the relationship between project uncertainty and success, for example, is one of many approaches that have been proposed as a solution (Aladwani 2002).

Part of the difficulty surrounding software development is exacerbated by the broad spectrum of software applications which range from embedded code in hardware to custom designed systems intended for specific purposes. The number of unknown variables that need to be taken into account differs from development project to project and introduces a high level of uncertainty into the planning and management of budget and schedule parameters. Other variables such as human, technical, environmental and political factors also have an effect and so the estimation of software projects will never be an exact science (Morgan 1994). Once uncertainty exists, risk is introduced, which project managers then need to manage throughout the life of the project.

Software estimation is an integral part of the overall development process as it represents many parts of the overall plan of the whole project and consists of critical components such as timeframes, resources and costs. The software estimation step or process within a development project has the ability to strongly influence the success or failure of a project and therefore requires much consideration and the appropriate approach. Considering that development projects are generally medium to long term the planning horizon is extended and the estimation task is made far more difficult. Estimation techniques also need to account for the customer or end user's expectations as far as functionality is concerned and technical complexity further complicates the accuracy of the estimation. Amongst all of these factors exists the need for effective communication between the project members and broader stakeholders to ensure what is promised is delivered. Lastly, there are both formal and informal estimation techniques which have been developed but need to continually adapt to the pace of change of technology.

1.3 Problem statement

This research aimed to establish how South African practitioners are approaching the challenge of software development estimation when dealing with bespoke software development projects.

1.4 Significance of the study

The study contributes to a better understanding of software estimation in a South African context. The research presents some of the latest thinking in the area of software estimation and will be of value to the software vendor and customer alike. As the dynamics of creating customised software in 'real time' for the customer are far more interesting and challenging when dealing with the translation between software functionality and customer requirements or expectations, the research looked at bespoke or customised software.

This study focused on published current findings and draws a comparison with the research findings obtained locally, and in doing so, highlighting areas that need to be improved and/or adapted to suit our environment. The intention is to provide software developers, architects, suppliers as well as IT managers and Chief Information Officer's (CIO's) with an insight into approaches to software estimation which can ensure successful development projects. Economically knowledge gained in this process will improve efficiency and productivity for the vendors and possibly even enable them to compete globally. Improvement in software estimation in turn will enable business to place greater reliance on the software development project parameters with the ultimate goal of delivering a successful product.

1.5 Delimitations of the study

- Software development of firmware was not addressed
- Enterprise Resource Planning (ERP) software was avoided

- The research focused on bespoke(customised) software developed for businesses in an attempt to improve their systems, processes, efficiency or other metrics that improved their competitive advantage
- The aim is to research software development projects in excess of one million rand although the prevalence of these projects has not been assessed
- The analysis of technical and functional design of software, although important, was limited to the point at which the end user and surrounding stakeholders were involved in formulating these designs
- Particular methods of estimation both scientific and informal were not analysed in any depth as the intention was to gain an opinion of these methods rather than an in depth understanding
- Respondents were restricted to the key stakeholders in relation to software development projects and estimation, specifically software professionals, developers and architects, software project managers and consultants.
- In all instances the influence and involvement of hardware in a software project was discounted as a factor. This is consistent with the secondary research which does not introduce hardware as part of the discussion.

1.6 Definition of terms

Table 1: Terms and Meanings used in the Report

<i>Term</i>	<i>Meaning</i>
Code	Program instructions written in a programming language by a programmer, also referred to as Source Code. Also executable machine code – source code that the computer can understand (MSN Encarta 2007)
Firmware	Software routines (embedded code) stored in read only memory (ROM) (MSN Encarta 2007)
Function Point	Basic function points are categorized into five groups: outputs, inquiries, inputs, files, and Interfaces. A function point is defined as one end-user business function, such as a query for an input. (Carnegie Mellon Software Engineering Institute 2007)
Information Systems	Collection or combination of programs, procedures, data, and equipment used in processing information. (MSN Encarta 2007)

1.7 Assumptions

- The key risk areas in the development process are well defined and common to the majority of software development.
- Traditional software best practices and processes have already been established and will cater for the majority of the theory of controlling budget and schedule.

- Due to the nature of software estimation, the South African context was not expected to be very different technically, although culture may be an influencing factor in determining success or failure rates.
- All software development is project based and therefore requires project management to be applied.
- Managing the cost of software development enables software companies to maximise the profit of their product, conversely exceeding cost to the client implies that the software vendor has also exceeded its original costs.
- Candid responses regarding the failure of software projects as a result of estimation or other factors may be difficult to extract as a result of human nature and organisational pride.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

There is a substantial body of knowledge around software development and estimation. In an attempt to extract the salient points from the extent of the secondary research that exists, the literature review discusses two positioning themes to the research topic and finally the central theme under the following main headings:

- Key Elements of Managing Software Development Projects
- Project Management of Software Development
- Software Estimation

As far as possible each theme is dealt with in a similar fashion, firstly by defining it, then by discussing the problems encountered within each area and lastly drawing out academic findings that may contribute to improving the overall body of knowledge of software estimation. The review concludes by integrating the three main themes and describing how each of the themes is positioned in relation to the problem statement. Four propositions arise from the secondary research and are reiterated at the end of the review with their relevant academic references in the order that they are originally stated during the course of the review.

A mind map precedes further discussion in order to provide a diagrammatical perspective of the literature review.

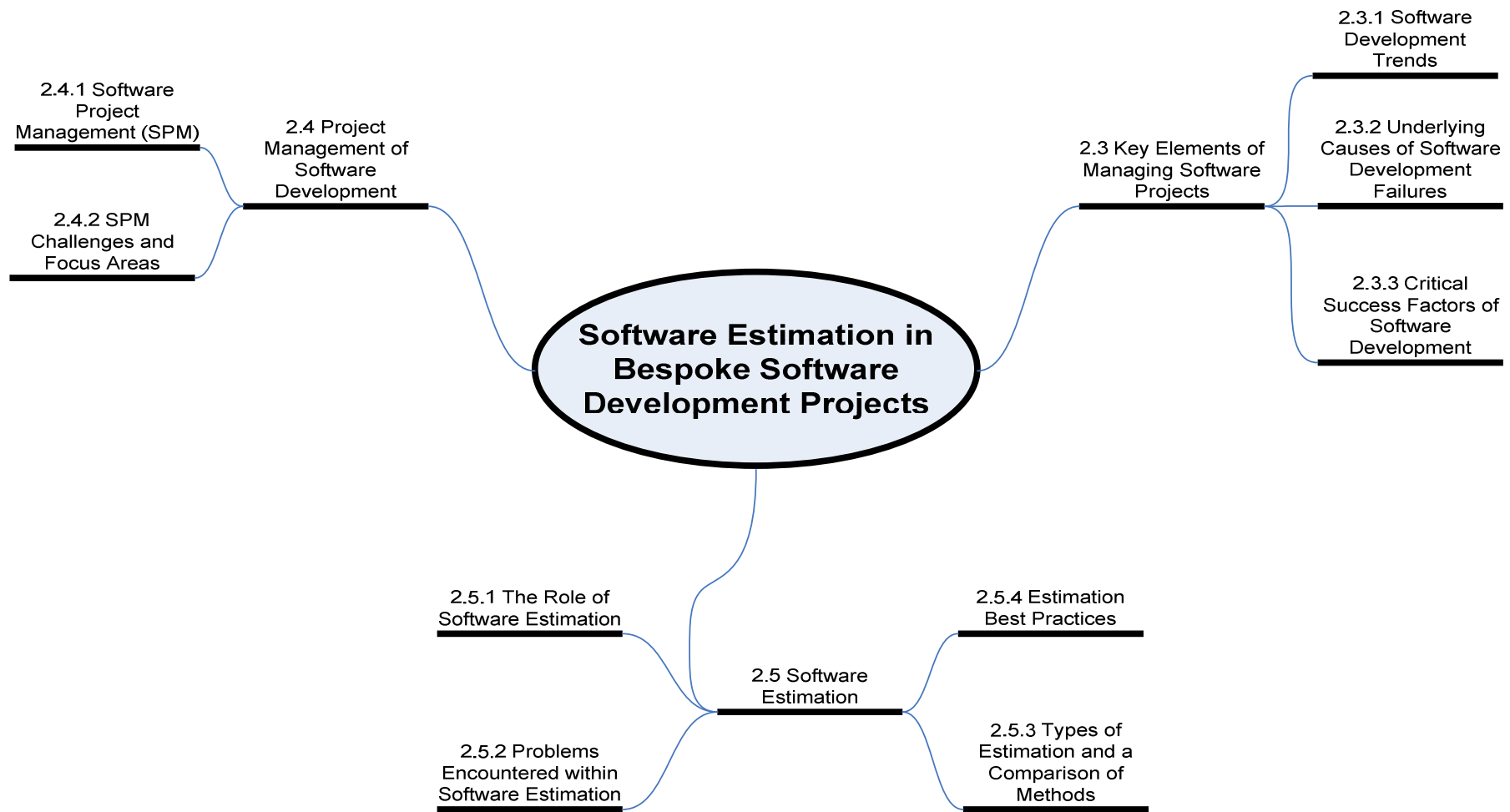


Figure 1: Diagrammatic View of Literature Topics and Sub-Topics

2.2 Background discussion

In order to resolve the status of unsuccessful development projects much research has looked at trying to find the root cause of their high failure rate, this task on its own is difficult due to the number of different types and the various sizes of software development projects that are undertaken. In addition the many facets of a software development make isolating a specific area problematic as the fault could be any number or combination of factors.

The majority of software development is undertaken on a project basis and therefore project management has a key role in managing the whole process from beginning to end. There have been substantial findings which have enabled areas in the software development process to be identified as critical to its success. From these areas, methodologies, models and approaches have been constructed in an attempt to provide an answer.

Software estimation forms part of the entire development process and plays a vital role right at the inception of all projects, until values can be ascribed to activities, resources and time the 'where, how, what and when' cannot be answered and an approach to the project difficult to formulate. In one authors view software estimation includes the estimation of development schedules, productivity and quality (Lewis 2001), while broadly accurate there are many finer points involved in software estimation that are discussed later in the review.

Lastly, the process of estimation itself has many approaches and there appears to be no single preferred method, at best there are approaches that suit certain projects.

2.3 Key Elements of Managing Software Development Projects

Cobb's Paradox:

"We know why projects fail, we know how to prevent their failure - - so why do they still fail?" (Cobb 1995:1)

In order to manage there needs to be an understanding of what is being managed. For this reason many researches and academics have attempted to identify what the reasons for the project failure was before they could develop any theory on how to manage it (Standish Group 1994; Whittaker 1999). By taking a look at the incidences of failed or unsuccessful software projects we may be able to determine whether or not the prevalence of failures is acceptable or not.

2.3.1 Software Development Trends

Based on the original CHAOS Report, research undertaken by KPMG in 1997 of unsuccessful technology projects in Canada, revealed that the majority of projects failed due to being over schedule, however, there was a high correlation between projects that were over budget and over schedule. 91% of projects that overran budget were also over schedule. Custom developed applications were the highest of all software projects that were both over budget and schedule at 69% (Whittaker 1999). In Whittaker's research a total of 61% of all respondents reported a failed project.

In later reports released by the Standish Group in 2001, as cited by Gowan and Mathiew (2005), the statistics had improved marginally but not to a degree that indicated that the problems were close to being resolved, the findings were that 28% of projects were on time and schedule but on average 45% of projects were over their cost estimates and 63% were over the original time to completion. Three years later Keil, Paul and Mähring cited the 2004 Standish Group results broadly, mentioning that more than half of software projects still experienced severe difficulties and/or failure.

Separate research undertaken by Jones (2004) which assessed 250 large software projects (10 000 or more function points) between the years 1994 to 2004 had empirically worse findings, only 10% of the projects were successful meeting schedule, cost and quality criteria. 175 of the projects experienced major delays, overruns or were not completed. These research findings strongly suggest that software development failure presents a major issue to the industry and that we appear to be making little progress towards resolving the key issues.

2.3.2 Underlying Causes of Software Development Failure

The software development environment is very complex and described as a 'network of causal relationships' by Abdel-Hamid and Madnick in 1989 and this remains true today. The number of variables present in a software project contributes to its complexity; moreover the combining of the human and technical element of the project is problematic and is certain to vary from project to project (Morgan 1994). The demand on software to adapt to and enable the rapidly developing hardware market creates an environment of continuous change and this affects both software quality as well as developer productivity (Quintas 1994).

In Whittaker's (1999) research the three main reasons for project failure were:

1. Poor Project Planning
2. A Weak Business Case
3. Lack of Top Management Involvement & Support

In Jones's (2004) study of 250 large software projects he found six common factors:

1. Poor Project Planning
2. Poor Cost Estimation
3. Poor Measurements

4. Poor Milestone Tracking

5. Poor Change Control

6. Poor Quality Control

It is important to note that all of the above six items relate to project management rather than technical involvement, this sentiment is supported by McManus (2004) who observed that software projects fail more often for management reasons than technical reasons. In Day's (2000) view the 'Software Crisis' as viewed is a result of the gap between IT and business, the fact that business has allowed IT professionals to continue to control the development environment is one of the main reasons that software development remains a quagmire. In addition McManus (2004) refers to direct and indirect stakeholders of whom direct stakeholders are core and argues that software projects can fail simply as a result of the end user or customer not accepting the new system.

The absence of quality within the software development process as already observed by Jones (2004) has also led to project failure, considering the aspect of continuous change in conjunction with poor quality, these factors can be directly responsible for increased costs, particularly after delivery (Tsoi 1998).

2.3.3 Critical Success Factors of Software Development

As established, two of the main contributing parties to the software development process are the technical team and the management team, these parties need to work in conjunction with each other for project success. The technical team essentially write the code and literally create the intellectual property (IP), this process is not according to a formulae which makes it important that the outcome is managed correctly in context with the user's requirements (Quintas 1994). Logically, deviation from the end user's requirements during this process will result in rework which impacts on schedule and costs in most instances.

The technical contingent have attempted to solve the software dilemma in their own way using pure mathematical approaches, models and tools such as

Computer-aided Software Engineering (CASE) to assist them in the process (Morgan 1994; Day 2000). These types of approaches have limitations and often are not able to adapt to the uniqueness of every software project (Morgan 1994). Kautz and Nielsen (2004) go further to say that technological approaches have had limited success, initiatives such as Software Process Improvement (SPI) have been developed as a solution to software failure but that the implementation in the organisation has not always been successful. While there is still advancement in this area it has already been identified that the management of software development is key and it is this area that will be focused on.

The Japanese were the first to introduce more formal management control over the software development process in the 1990's, they also were able to reduce the costs of producing the software by introducing quality programs (Quintas 1994). Kautz and Nielsen (2004) also found that the implementation of quality management is often unsuccessful in an organisation although Jones (2008:3) advocates that "Effective software quality control is the most important single factor that separates successful projects from delays and disasters." In general the earlier problems are detected the less costly they are to fix. It is also acknowledged that software development is not error free and this must be allowed for in the development process although quality overall will translate into increased customer satisfaction. Quality should therefore be built into the process of constructing software (Shumate and Snyder 1994; Barnett and Raja 1995).

In research regarding projects that were completed on time, a formal project methodology was determined to be imperative to their success irrespective of complexity or size. The empirical research proved positive relationships between project methodology and project performance as well as the ability to be on schedule (Gowan & Mathieu 2005). The authors emphasize that the methodology must be tailored to the project and exist throughout the life of the project, although the authors do not refer to project management in any official form it assumed that this would be the PMBOK® methodology by default.

2.3.4 Conclusion to Section 2.3

In summary the literature suggests that the failure of software projects is prolific and this is well supported by empirical data. There is certainly validity in the literature presented although insufficient current data that enables a 'real time' perspective. Further discussion in the literature review however, supports the idea that currently the issue of software development failure continues to exist.

Software projects fail for a number of reasons; the projects themselves are complex and require input from many sources. As a result the coordination of software projects is difficult as they are controlled by both technical and organisational sources which in turn have their own dynamics. Continuous change in the environment has made the task of formalising a specific approach to software development projects even more difficult as well as the implementation of these approaches.

There are technical and managerial approaches that have been put forward to control these projects and this section of the review focused primarily on the managerial aspects in support of the secondary research cited. Findings support that a formal project methodology, conceptualised as project management, should be adopted irrespective of project size, components like planning and quality are important contributors to a projects success as well as stakeholder involvement.

2.4 Project Management of Software Development

Prior to any software being used operationally it first has to be developed which involves a number of integral steps which all need to be coordinated in a logical sequence. It makes sense therefore to deal with these activities as a project, making use of a cross functional team linked by a common overriding objective and set of deliverables.

The role of project management in this realm is therefore critical and has existed as a discipline for many years. The composition of project management is probably best encapsulated in the PMBOK® *Guide* (2000) which articulates in

detail the pillars of traditional project management. Taking into account the complexities of software development already discussed, project management provides a framework to plan, coordinate, manage and control the software development process while addressing the critical areas such as quality and stakeholder management of the project.

2.4.1 Software Project Management (SPM)

A key finding in the research by Whittaker was that the lack of execution of project management was ranked as the number one finding when analysing budget and schedule overruns (Whittaker 1999). Referring back to the 2001 Standish Group figures, the report proceeds to advocate making use of “skilled project managers” and “good project management practices” to improve the performance of Information Technology (IT) projects. The research by Gowan and Mathieu (2005) emphasizes the application of a formal project methodology irrespective of the scale or complexity of the project and although this research focused on completion date, it ties in with the earlier findings of Whittaker.

Software Project Management (SPM) is viewed by many as a more comprehensive approach (Nienaber and Cloete 2003) which has evolved and been adapted from the nine traditional knowledge areas of project management as it is unique due to the nature of software development. According to Nienaber and Cloete the three key elements of SPM are:

1. Project Stakeholders
2. Project Management Knowledge Areas
3. Project Management Tools and Techniques

In Nienaber and Cloete’s view, SPM involves all the aspects of software development being:

1. Scope and Objective Identification
2. Planning

3. Evaluation
4. Project Development Approaches
5. Software Effort and Cost Estimation
6. Activity Planning
7. Monitoring and Control
8. Risk Management
9. Resource Allocation and Control
10. Managing Contracts
11. HR
12. Quality

Interestingly McManus (2004) differs in opinion slightly and separates project management from Software Engineering and Quality Engineering, the three together he believes make up the dimensions of software projects. What is evident is that the use of all the facets of project management may not always be applicable depending on the scale of the project but it is important that all the formal procedures are present. The absence of formal procedure causes informal procedures to become established which may not fulfil the requirements of the task and as such there may be a break down in the management of that task (Robinson and Ringer 1999).

2.4.2 SPM Challenges and Focus Areas

Project success is ultimately the goal of project management and project managers, although the definitions of success vary it is reasonable to accept that a software project is successful when “actual outcome matches or exceeds expected outcome” (Ross 2003:3). In order to achieve this success the project manager must continually ensure that the actual and the expected correspond,

if they don't then he or she must take the necessary action to reconnect the two as expediently as possibly (Ross 2003).

Software project managers have a primary responsibility to manage the inherent risks of a project although it would seem that few project managers and academics including Boehm, Jiang and Klein can all agree on the most important of these risks. It would appear that although more experienced project managers perceive more project risks than less experienced project managers, no group including the abovementioned academics have a single, main, common risk factor (Smith, Eastcroft, Mahmood and Rode 2006). Having ascertained this Smith et al (2006) identified four risk factors that were unique to South Africa which did not appear in the secondary research they had undertaken, they ascribed their findings to a unique organisational culture as well as a less mature software industry as opposed to Europe and the US.

These findings highlight perhaps why there is no main common risk factor but do not deny that risks within a project must be managed on many fronts. Quality, cost and schedule all represent large areas for project management, Jones (2004) found that poor quality was the primary contributor to cost and schedule overruns and secondly that poor project management was the main cause of unsatisfactory quality. Other research identified four out of the twelve independent variables used had as much as a 52% cumulative effect on cost reduction where 'cost reduction' was set as the dependant variable (Javed, Manzil-e-maqsood and Durrani 2006).

These variables were:

1. Common understanding of project domain and requirements definition (17%)
2. Managing client change requests (11%)
3. Managing delays caused by the client (11%)
4. Understanding and meeting the clients definition and software quality needs (13%)

Managing the project schedule through planning and control is another risk area that has pitfalls which are exacerbated by project managers who fail to report the actual status of the project in order to protect themselves. The accurate status of a project is an important control mechanism and non-reporting has a great risk and also been found to be more prolific in certain cultures (Javed et al 2006, Keil, Paul and Mähring 2007).

Jiang and Klein (2001) in looking at software project risks found that fulfilling the user and stakeholders' requirements was the greatest obstacle for information systems project managers. McManus (2004) believes that SPM literature does not clarify the stakeholders and strategies to deal with them during the project, McManus feels that project managers need to have experience with stakeholders and amongst other things make use of "consensus-based" approach for deciding on the objectives and deliverables of the project. Gallivan and Keil (2003) advocate user participation but caution that their involvement is not a guarantee of project success, similar to McManus (2004) they believe in open communication as well as understanding the different stakeholders better in order to ascertain their influence and impact on the project. Ultimately the project must deliver according to the user's needs to attain project success as per Ross's (2003) definition.

2.4.3 Conclusion to Section 2.4

It is clear from the literature why project management is well suited to cope with all the components of software development. Similarly applying SPM correctly is challenging and can also lead to the demise of a project if done incorrectly. Nevertheless an approach does exist that is suitably equipped to manage the complexity of software projects, it is the nature of the project that requires more or less focus in some or all of the areas identified.

Review of the literature discussed indicates that there are protagonists for focus by project managers on certain critical areas and evidence suggests that not all areas in the project management realm are of equal importance. Software project management is a human intervention and therefore not an infallible process, managing the risk elements of the project effectively will have far

reaching effects due to the interrelated nature of factors like quality, cost and schedule. Similarly the absence of controls in these and other aspects of the project will greatly reduce its chances of success.

User involvement remains a common theme although stakeholders need to be managed and communicated with on varying levels based on their involvement in the project. In maintaining consistency with the definition of a projects success it is largely the adoption by the user of the software that will dictate its success.

2.5 Software Estimation

In Nienaber and Cloete's (2003) list of the twelve aspects of SPM they list "Software Effort and Cost Estimation" in their list, the literature review has already discussed at some length the shortfalls of software development and the reasons for it, to highlight one reason according to Ross (2003), project failure occurs because of unreasonably high expectations. Research by Tsoi (1998) found that 'good' effort estimation was the most important reason for successful software projects. These two statements are related by the fact that software estimation attempts to gauge the size and complexity of a project in order to arrive at the project's cost, effort and timeframe requirements at the earliest opportunity (Tsoi 1998).

Based on estimation, expectations relating to a project are formed, due to the pervasiveness of software applications in our everyday lives, Lewis (2001) believes that software professionals have a duty to make estimation as honest and transparent as possible. The author feels that inflated and unrealistic estimates are damaging the reputation of the software industry.

2.5.1 The Role of Software Estimation

Software size has the greatest effect on the effort, cost, manpower, defects and development schedule of a project (Ross 2003), project scope incorporates software size and the size of the project typically measured in, effort, cost and features or functionality (McConnell 2006). Software estimation is a generic

term predominantly relating to effort estimation and cost estimation, its role is to “establish effort, timeframe and ultimately the cost of delivering a prospective project” determined by the project scope (David 2006:1).

The purpose of software estimation is for the benefit of both the software provider and the acquirer of the developed software. There are four main purposes; firstly it allows for budgeting where accuracy is vital, secondly, estimation enables risk analysis by looking at cost and schedule tradeoffs. Thirdly, planning and control in order to ascertain cost components and timelines. Lastly, estimation looks at software improvement and investment analysis to evaluate various costs and benefits (Boehm, Abts and Chulani 2000, Jorgensen 2007).

2.5.2 Problems Encountered within Software Estimation

In fulfilling its role, software estimation has a number of challenges to overcome in arriving at a reliable and accurate estimation. The project team need to be able to rely on it for planning and the customer to assess costs. Dekkers (2005) lists the following points as the primary challenges of estimation:

1. Accuracy of input values (size, complexity, technical requirements)
2. Availability of input variables
3. Ability to apply historical databases to the new project
4. Completeness of the requirements (Functional, Non-functional and Technical)
5. Tasks that need to be included
6. Risk factors

According to Ross (2003) software doesn't have clear dimensions that can be measured. In his view software is fluid and takes form over the period of the project. When setting out to develop software, Ross believes that some of the most critical decisions are made when the least is known and therefore at a

point where uncertainty is greatest. McConnell (2006) supports Ross's viewpoint of uncertainty, he refers to project variables which are not clearly defined at the start of a project and the variability that these create in the estimation task. McConnell (2006:1) encapsulates the variability of a project within a "phenomenon known as the Cone of Uncertainty" illustrated in Figure 2 below:

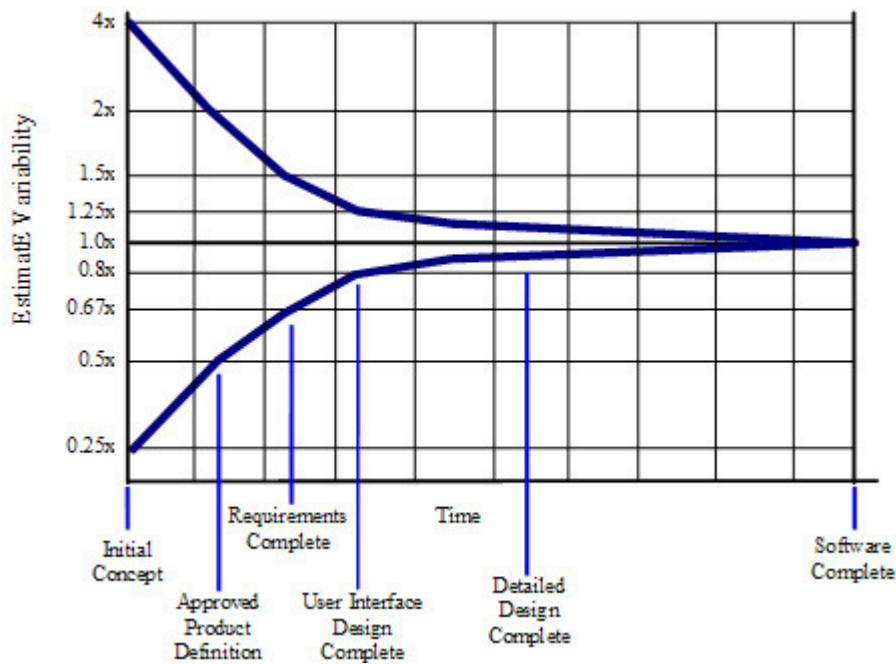


Figure 2: The Cone of Uncertainty (McConnell 2006)

Based on the concept of the Cone of Uncertainty software companies make undertakings too early in the cone and the result is that they weaken the predictability of the estimate, increase risk and compromise the success of the project (McConnell 2006). In parallel to the Cone of Uncertainty is the influence of the project stakeholders at the inception of the project, according to Javed, Manzil-e-maqsood and Durrani (2006), the stakeholders have the greatest power to determine the end product and project costs, as the project progresses their influence reduces accordingly. Considering the power the stakeholders hold at this point, one assumes an additional burden to the complexity of an accurate estimate. Lewis (2001) interprets his research findings far less scientifically and states that software development cannot be impartially estimated and remains subject to opinion.

The concept of 'Project Scope' has already been discussed and the role of estimation in estimating the projects scope, this task is viewed as the most difficult element of software estimation by Roetzheim (2005). Armour (2004) justifies this viewpoint in saying that when we attempt to measure software size, a key element of a projects scope, we are in effect attempting to measure knowledge which cannot be done empirically. Incorrect estimation of project size is often due to the nature of the sizing approach according to Jones (2004) which may understate the size of the work contained in the project. Armour (2004) debates the use of effort, measured in staff hours or 'person' months, to size a system as opposed to function points or source lines of code (SLOC) indicating that there alternative views in the approach to size estimation. Later on though, Armour (2004) states that using effort to measure software size may be flawed due to the amount of unseen effort that would be problematic to account for.

From more of a business perspective, a sample of software litigation cases researched found that the primary cause for litigation was the absence of an accurate estimate or rejection of the estimate (Jones 2008). Of the cases mentioned, half had competent estimates but were replaced by estimates that suited the organisations needs and not the project team's capabilities. Javed et Al (2006) found a customer's understanding of software estimation to be deficient when concerning size, cost and effort, particularly when wanting to introduce additions later in a project which resulted in increased costs.

Software companies have been found to provide "over optimistic effort estimates" according to Jorgensen (2007:1) in an attempt to secure business, a second distortion that Jorgenson found was overconfidence which also resulted in a lower estimate. Stellman and Greene (2005) found that estimates were invariably too high or too low, similar to Jorgenson estimates that were too low were as a result of overconfidence, an aggressive stance by management or unrealistic goals. Those estimates that were too high, commonly referred to as 'padding', were created by project teams that tried to incorporate additional time into the project as a buffer. A factor of these types of estimates was revealed to be the influence of irrelevant or misleading information by Jorgenson and

Grimstad (2008), the authors argue that all information not relevant to the requirements of fulfilling the project itself, for example customer cost expectations, should be filtered out irrespective whether a judgement or model based approach is used in creating the estimate.

2.5.3 Types of Software Estimation and a Comparison of Methods

There are a number of different approaches to software estimation all of which need to fulfil the role of estimation. The objective of estimation models is to map out a clear development schedule and provide an accurate cost estimate to produce the software product (Boehm, Abts and Chulani 2000). Figure 3 illustrates the various estimation techniques according to Boehm et Al (2000) however, only Model-Based and Expertise or Expert-Based will be discussed within this review as they have been found to attract the most comment and represent the two most common opposing methods.

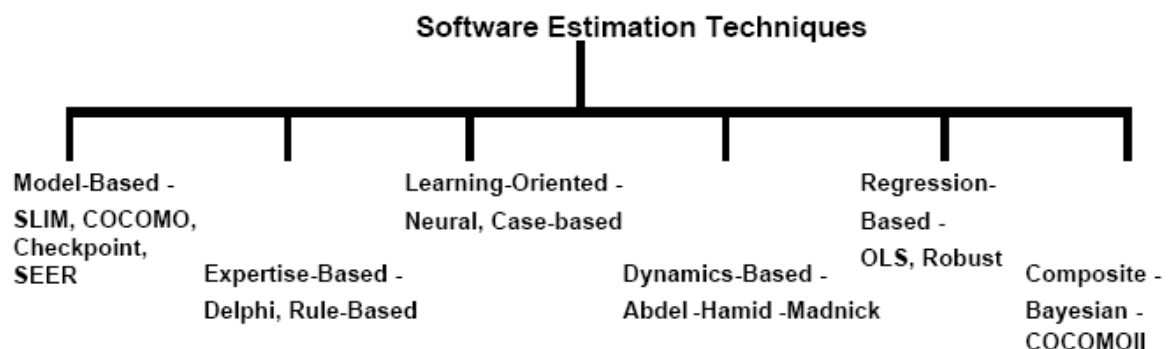


Figure 3: Software Estimation Techniques (Boehm, Abts and Chulani 2000)

According to Jorgensen (2007), the difference between Expertise-Based and Model-Based techniques is the final step before an effort estimate is produced or otherwise termed the 'quantification' step. If the final step is judgement based then the technique is expert based if the final step is mechanical then the technique is model based. This differentiation is not sufficient for the purposes of this review and greater detail of both techniques will be discussed further on.

A point to note is that the term 'judgment' is often used interchangeable by academics with 'expert' and will be done so within this review as well.

Expert (Expertise-Based) estimation is defined broadly by Jorgensen (2004) as a method that exists between unstructured intuition and expert judgement, the latter based on historical data, procedures and structure. The author claims that this method of estimation is the most prominent of the techniques when assessing effort. Importantly the term 'expert' must be put into context, Jorgensen considers software professionals, project managers and software practitioners as such. In addition the amount of knowledge the expert has applied to the project must also be taken into account. Stellman and Greene (2005) imply that they had similar findings, the estimates they encountered that were based on previous experience proved to be the most precise of the estimates.

Jones (2005) however, when looking at large projects found what he termed as 'manual' methods to be inaccurate and overly inflated whereas the 'automated' estimates were more accurate. The model based method was better equipped to deal with the complexity of larger projects and faired far better when assessing the non-coding or programming aspects of the project. According to Jones (2005) successful projects made use of at least one of the multitude of software cost estimating tools on the market.

For ease of comparison, Table 2 lists the respective advantages and disadvantages of Model-Based estimation versus Expertise-Based estimation as per Boehm et Al. (2000) and Jorgensen (2004, 2007).

Table 2: Advantages and Disadvantages of Model and Expertise-Based Techniques

Model Based	Advantage	Disadvantage
	1. Good for budgeting, trade off analysis, planning and control and investment analysis when calibrated on past experience	1. Accuracy is compromised when dealing with unprecedented situations, pace of software development has made it difficult to maintain high accuracy
	2. Where little or no domain knowledge is available more likely to create accurate estimate	2. Few organisations fully understand workings of models and are not comfortable using them
	3. Removes human bias where strong personal interest exists in project outcome	3. Reliant on correct calibration relative to the context of the situation, difficult to translate context though
Expertise-Based	Advantage	Disadvantage
	1. Useful in absence of empirical data, experts deemed to have a natural advantage	1. Estimate is equivalent to the experts opinion, no measurement available until consequences occur
	2. Able to capture knowledge and experience of experts in that domain	2. Years of experience do not equal high competency
		3. Competent individuals can still make errors in judgement

A closer look at only two of possibly the most widely used techniques reveals that no single approach satisfies all the objectives of software estimation. In the writer's opinion and experience, no other estimation technique fully addresses

all the requirements of software estimation either and therefore there would be similar findings if all the techniques were assessed.

2.5.4 Estimation Best Practices

Jorgensen (2004) reviewed twelve expert estimation 'best practices' and found justification for all twelve of the guidelines. Of these guidelines the following were found to be independently supported by other academics.

1. **Avoid conflicting estimation goals** – customers invariably are requesting a price for the software development rather than an idea of what effort the project requires. In order to provide an accurate estimate, practitioners should first create the effort estimate without any consideration of price (Jorgensen and Grimstad 2008)
2. **Estimators must justify and criticise their estimates** – those responsible for creating estimates are pressurised by the client and management to provide an optimistic or aggressive estimate in this case the estimator must be able to defend the estimate. A reliable defence is an estimate based on the historical data of like projects (Jones 2005)
3. **Estimate top-down and bottom-up** – Good estimates should ensure that the customer needs are met; estimators must establish the size and complexity of the customer's need. The project team must determine from their side how difficult it will be to fulfil the customer's need and specify a confidence level to the project (Dekkers 2005).
4. **Use documented data from previous development tasks** – Tsoi (1998) advocated that estimation knowledge was about building a historical database to allow for improvement and the reuse of software design and components, the term the author used was a "planning experience transfer".
5. **Find estimation experts with relevant domain background and good track records** - In an attempt to address the inefficiencies of model-based techniques Jones (2005) believed that knowing how to adapt the

estimation tool to the respective conditions was the answer. Jorgensen (2007) agrees and states that the estimation model should be 'calibrated' to fit the organisation, which could then lead to better results. One method of calibration would be to use historical data specific to the organisation. In a similar but more general vein, Sheppard and Kadoda (2001) established that one should rather identify a 'prediction system' that matched the context as opposed to opting for the best system.

6. Combine Estimates from different experts and estimation strategies

– Jones (2005) in his own version of a best practice supports the use of a combination of estimation techniques administered by software experts. The concept of making use of more than one estimation method is also a recommendation of Boehm, Abts and Chulani (2000), they refer to the combination of techniques as 'composite' and the point is not to favour one technique over another but to determine why one estimate produces a marked difference from the other. Boehm et Al. do state however, that all the techniques are being outpaced by the changes in software technology.

7. Assess the uncertainty of the estimate – McConnell's (2006) 'Cone of Uncertainty' clearly illustrates this point, Ross (2003) and Dekkers (2005) concur and emphasise that uncertainty changes over time, customers must be advised that for this reason the estimate cannot remain fixed. Ross (2003) proposes that estimates should be represented as a probability distribution rather than one value.

8. Provide estimation training opportunities – Often the acquirers of the new system do not understand the reason why a project may take that length of time but at the same time their acceptance is necessary. Stellman and Greene (2005) advise that the project manager must communicate and explain the estimation process to the stakeholders in order for them to have insight into the project. Where an element of distrust exists between the technical and non-technical teams, due to a lack of knowledge on the latter's behalf, standardised practises go a long

way towards helping all stakeholders trust and understand the processes involved.

2.5.5 Conclusion to Section 2.5

Of the three most important practices that lead to software project success, planning and estimation, before the start of the project is one of them (Jones 2004). Having said that, review of the effort and cost estimation process has a defined purpose in the software development realm, however, there are a number of complexities involved in actually producing a realistic and accurate estimate. Furthermore there are a number of approaches that are employed when producing an estimate none of which are clearly the most reliable or robust method. Within this review there appears to be two main schools of thought, one which the favour model or automated estimation method and the other that favour the judgement or expert-based approach. These two methods are not the only methods but do appear to be the most prominent among those researched.

A great deal of consideration is given to the level of uncertainty pertaining to key variables of the project, software size, effort, timeline and cost all need to be determined in producing a software estimate. Estimating these variables at the start of the project when the level of uncertainty is highest suggests that they are likely to change as the project progresses. The variables should therefore be expressed within a probability range rather than absolute values. The relationship between uncertainty and risk that software experts need to manage is complicated further by the different interests and expectations of the stakeholders. Estimators are pressured by cost constraints and aggressive development schedules as well as unrealistic functionality expectations all of which lead to estimates that are either too high or too low.

In a quest to improve the know-how of software estimation, academics have put forward best practices and guidelines which can be used as a foundation towards creating accurate estimates. One of the main underlying themes is that software projects are seldom identical and neither are the environments that they are created for and this leads to certain techniques suiting certain

situations and other instances where best practices advocate making use of a combination of techniques.

Proposition 1: Expert estimation techniques are preferred to model-based techniques in South Africa

Proposition 2: Business interests as opposed to project scope are the greatest influence in arriving at software estimates

Proposition 3: Estimate values are committed to at the start of the project and are seldom revised

Proposition 4: Despite a combination of estimation techniques being advocated by academics this is seldom practised in reality

2.6 Conclusion of Literature Review

The literature regarding software development is substantial and varied; understandably an industry of this size attracts a great deal of attention to it. The enigma of successful software development persists, in particular the inability to meet budget or schedule constraints. Justifiably there are many viewpoints; the nature of software development dictates there may be a number of solutions but there are, however, a few main schools of thought around solving the issues. In the South African context there appears to be little statistical data and even internationally the research is largely qualitative. Arguably the task of gathering such data is particularly onerous given that the issue is not one that companies would like to portray. Solutions do exist but there is certainly no generic answer (Morgan 1994), in the authors opinion change, human interaction and technical complexity are key to the topic.

Of primary interest within this literature review was the non-technical aspect of software development, the management practices that are employed to orchestrate the entire process and deliver successful projects. Software project management was discussed at length as being an important factor in facilitating software projects. The role of SPM is challenged in its responsibility to bring together the technical and non-technical aspect of the projects, manage

stakeholders, contain costs, ensure quality, manage risk, meet client expectations and be on schedule. SPM like traditional project management has a number of identified focus areas which enable it to achieve project success. Software estimation is one of the twelve areas and its role is of primary importance particularly at the outset of the project.

The term 'estimate' suggests that the value or values arrived at are based on what we know now and may be likely to change as our knowledge of what we are estimating grows. The estimate also serves as a basis for the investment decision by the client and planning for the project manager and team based on the project size or scope. As a great deal of decision making takes place at this point where uncertainty is greatest, the estimate should be as accurate as possible.

There are many ways to arrive at an estimate for a software project all with advantages and disadvantages, some that are better suited to certain types of projects than others. In particular model and expertise-based methods featured in the review, both of which were well supported by various academics and business practices. In addition, academics of these techniques released findings that a combination of methods was a more holistic approach to take. Irrespective of the method applied estimates are very often too high or too low, perhaps an obvious finding, however, it is the reasons behind an inflated or overly optimistic estimate that are important. Different stakeholders are able to influence the estimate based on their interest in the project, this reduces the probability of an accurate estimate and increases the chance of an unsuccessful development project.

Propositions to be tested:

2.6.1 Proposition 1: Expert estimation techniques are preferred to model-based techniques in South Africa

(Lewis 2001; Jorgensen 2004; Stellman and Greene 2005)

2.6.2 Proposition 2: Business interests as opposed to project scope are the greatest influence in arriving at software estimates

(Stellman and Greene 2005; Jorgensen 2007; Jones 2008; Jorgensen and Grimstad 2008)

2.6.3 Proposition 3: Estimate values are committed to at the start of the project and are seldom revised

(Ross 2003; McConnell 2006)

2.6.4 Proposition 4: Despite a combination of estimation techniques being advocated by academics this is seldom practised in reality

(Boehm, Abts and Chulani 2000; Jorgensen 2004; Jones 2005)

CHAPTER 3: RESEARCH METHODOLOGY

Research methodology is necessary in order to map out the process of gathering the primary data in order to compare it to the documented sources already covered in the literature review. To a large extent methodologies are either quantitative or qualitative and from this point there are established routes that can be followed in order obtain the primary data. It is important that the research methodology is aligned and that it meets the end objective of providing rich data in order to extract a meaningful analysis.

The research instrument serves as the tool to obtain the raw data and must be constructed in such a way as to be as neutral as possible. It is accepted that every methodology has its weaknesses and these must be guarded against as far as possible when selecting a population and sampling method. In the writer's opinion, it is critical to present the analysed data as originally as possible in order for it to be of value to any party interested in its findings.

3.1 Research methodology /paradigm

The research method chosen was qualitative for various reasons, given the general absence of any literature on the research topic in South Africa the attitudes and perceptions of the sample are what assisted in establishing concurrence or variation with what had already been documented in the literature review. It is the underlying qualitative issues and factors that are initially of greatest interest in the South African context. Based on the qualitative findings, one can now choose to probe more accurately using numbers to test hypotheses that are either unique to South Africa or dissimilar to the findings in the secondary research herein.

The testing of proposition one to four are all justified on this basis, in all instances, the research was intended to be exploratory and the propositions tested in a South African context. This is potentially the starting point for further analysis given the general absence of this type of information in the country.

3.2 Research Design

The research was carried out as a semi-structured interview with an overriding philosophy of obtaining the software specialist's point of view. Firstly the semi-structured approach allowed for flexibility which was needed for this philosophy and as the research was exploratory this approach did not limit the discussion of the central ideas. The intention was that the same research instrument be used for all respondent types. Ongoing secondary research was undertaken where the findings required additional support and referencing.

The advantage of using the abovementioned approach was that the respondents were focused and qualified, representing expert and business decision maker's opinions. The method allowed for opinions to be expressed and exposed the researcher to personal insights of the respondents. The research could also be gathered in a constructive order i.e. the most prominent or qualified of respondents could be targeted first in order to obtain a comprehensive response to the questionnaire. This did not suggest that respondents had different weightings; the intention was to achieve a reliable 'base' early on in the collection of data.

The disadvantage of the design was possibly the semi-structured approach and the potential for the interview to evolve and become an iterative process depending on the relevance of the questions. This concern did not materialise and the research instrument remained unchanged after the pilot was carried out, it appeared that the content of the questions was sufficient to illicit sufficient responses from all the respondents. According Neville (2005) all interviews whether structured, semi, or unstructured have some problematic areas to be avoided by the interviewer:

1. Bias – from either the interviewer or interviewee based on various factors
2. Demeanour of the interviewer – Tone of the interviewer, appearance, interest in the topic, response to respondents answers and body language all play a part in influencing the interviewee's response
3. Suspicion of the interviewer – concerns about the motive for the interview

4. Conduct of interview – particularly at the start of the interview, the interviewer must be appreciative of opportunity and set the interviewee at ease regarding the format and expectations of the interview
5. Confidentiality – related to suspicion, interviewee's must feel confident that the information will remain confidential

3.3 Population and sample

3.3.1 Population

The population comprised of software vendors, suppliers, and professionals that operate primarily within the realm of software estimation, development and implementation. The population had ideally undertaken a software project in the last two years or were in the process of undertaking a software project no less than one million rand in value.

3.3.2 Sample and sampling method

The sample size was intended to be a maximum of twenty respondents. The software vendors, suppliers and professionals were segmented by their position in the value chain with the main focus on software estimation professionals, developers, project managers and implementers. The respondents were pre-qualified in order to establish that they were either involved in software projects and estimation or had been previously, they were asked to partake in the interview by way of a covering letter along with the interview content. The purpose of the interview was explained to the respondents and why they were approached as respondents. All respondents were extended full confidentiality, confirmed in writing, as well as a copy of the final report for their organisation once it is completed. Table 3 below provides a profile of the intended respondents and Table 4, thereafter lists the respondent names and organisations in order of interview date.

Table 3: Profile of Respondents

Description of Respondent Type	Number to be sampled
Software Estimation Professional	10
Software Developer	5
Software Project Manager	3
Software Implementer	2
Total	20

Table 4: Respondent Names and Organisations

Respondent Name	Organisation
1. Garry Ackerman	Galdon Data
2. Ken Everet	Digital Matter
3. Darren Beams	Digital Matter
4. Jimi Lutz	3 Fifteen
5. Brand Zietsman	3 Fifteen
6. Roy Hobson	Oakwel/ Silverminute
7. Hennie van Wyk	Silica Holdings
8. Caron Mooney	IS Partners
9. Nathan van den Berg	Britehouse
10. Ryno Loots	Value Analytics
11. Pamela Day	Meditech
12. Douglas Lunt	Ericsson – Sub Saharan Africa
13. Johann Strydom	SITA
14. Peter van der Vegte	iSolve Business Solutions
15. Bryan Maiden	VoxAmvia

3.4 The research instrument

The research instrument was a questionnaire consisting mainly of open ended questions designed to cater for all respondents. The questions were appropriate to each respondent type in terms of their phrasing; all respondents were asked the same questions although discussion around each topic or theme varied depending on the flow of the interview (Neville 2005). The questions in the

instrument were intended to guide the interview and illicit honest and considered responses from the candidates and therefore did not follow a strict interview schedule.

The questionnaire consisted of a General introductory section followed by four sections all focused on Software Estimation as follows:

- Software Estimation
 - A: The Role of Estimation and Estimation Techniques
 - B: Estimation Influences
 - C: Estimate Values
 - D: Estimation Best Practices

3.5 Procedure for data collection

The interviews were face to face with the respondents and all interviews were recorded to ensure that the primary data was captured accurately. In addition supplementary notes were taken by the interviewer and combined with any material that was created or provided by the interviewee that was non verbal, for example diagrams, notes, documentation that the interviewee provided in response to the questionnaire.

3.6 Data analysis and interpretation

All the responses were divided and captured in Excel format per respondent, thereafter each question was analysed for commonalities and trends that emerged. The intention was to map the concentration of answers of each question and then analyse corresponding trends between the different respondent groups. The expectation was that there would be trends or commonalities arising from the responses that were unique to each respondent type and a concentration of responses that were common to the respondent groups.

The sample size was small therefore consolidating the responses and identifying trends, or the lack thereof, was relatively straightforward, thereafter these findings were consolidated as described above. There is no apparent or stated methodology to this approach.

3.7 Limitations of the study

Challenges to the 'phenomenological' or qualitative approach, as per Neville (2005) have already been listed under 'Research Design', further to that the author cites the following limitations:

- Difficult to interpret findings
- The ability to collate findings in a meaningful way
- Findings are subjective and difficult to reproduce

Bell (2006) adds to the limitations of qualitative research in general and lists the following points:

- Sample size is relative and time available given the scope of the research project
- Method is a subjective technique and therefore susceptible to bias
- Findings are subjective and the ability to generalise them is limited by this

The highly diverse nature of software applications, notwithstanding customised applications, determines that the experiences of each software development project and therefore the estimation process will range in terms of the type, for example packaged versus custom developed applications, used in each software solution. The literature review however has no bias in favour of any software type or configuration and the findings could therefore be used as indicators for this research despite the sample variety and size.

3.8 Validity and reliability

Validity and reliability enable research to be both meaningful and extended by others in the writers opinion, put differently the two characteristics allow the research to 'carry weight'. Research that lacks these qualities does not make a valuable contribution and therefore has not addressed the primary purpose of research. The aspects of validity and reliability are related in the sense that research may be valid and unreliable but never unreliable and valid (Bell 2006).

3.8.1 External validity

External validity according to Neville (2005) is research that measures "what it is intended to measure", more broadly; can the research be applied more generally? In this instance the methodology adopted challenged the external validity as the population specified was segmented in terms of only four software professions and the sample was relatively small. The justification for this was based upon establishing specific trends from a focused sample rather than general trends from a homogenous population. The strength in this research therefore lies within the internal validity rather than the external validity.

3.8.2 Internal validity

According to the Validity and Reliability power point presentation (2007) internal validity measures whether the research examines the topic sufficiently. This research has a probabilistic (Neville 2005) sampling method specifically for the purpose of understanding the viewpoints of the software professional. The interview method allowed for depth in terms of insight although the sample size was limited by time and manpower constraints. Specifically the research approach selected was aimed at obtaining some definitive outcomes for further exploration.

The interviews were recorded allowing the option for them to be transcribed in order to restrict bias and be cited accurately.

3.8.3 Reliability

Bell (2006:117) describes reliability as research that obtains the same or similar results “under constant conditions”. Neville (2005:26) refers to research and reliability as research that produces “consistent findings”. This posed an obstacle for this research because by default the research approach was qualitative and searched for insight and opinions from individuals. These insights and opinions were likely to vary from person to person.

The sample group was, however, segmented and provided meaningful information pertinent to the population identified. The structuring of the questionnaire also tried to address reliability as far as possible without leading the respondents. This, in addition to performing a pilot of the interview prior to collection of the primary data, addressed the requirement of amending the questionnaire if needed. Finally the writer is employed in the software industry and although this may have had bias implications, he has sufficient experience to construct a robust questionnaire.

CHAPTER 4: PRESENTATION OF RESULTS

4.1 Introduction

The findings of the primary data obtained from the questionnaires are presented below using tables and figures to augment the associated text. In some instances, findings that are relevant to a particular proposition that do not warrant a chart or table are described in text. The presentation of the findings begins with the desired sample described in the research methodology chapter in comparison with the demographic profile of the respondents interviewed. Thereafter, the findings of each of the four propositions are presented in the same order as presented at the conclusion of Chapter 2. A description of all the findings is provided with each item of data displayed for each proposition discussed.

4.2 Demographic profile of respondents

Table 5: Profile of Intended Respondents

Description of Respondent Type	Number to be sampled
Software Estimation Professional	10
Software Developer	5
Software Project Manager	3
Software Implementer	2
Total	20

The population from which the sample in Table 4 originated had three parameters outlined in Chapter 3; firstly, they had to operate primarily within the realm of software estimation, development and implementation, secondly, the population had undertaken a software project in the last two years or were in the process of undertaking a software project, lastly the project was ideally no less than one million rand in value. Overriding these parameters is the focus of the thesis around bespoke software development as bulleted under the delimitations of the thesis in Chapter 1.

Table 6: Skills Matrix for Respondents Interviewed

Respondent	Estimation Professional	Developer	Implementer	Project Manager
1	X			
2		X		X
3	X			
4				X
5				X
6				X
7	X			
8	X	X	X	
9		X	X	X
10	X	X	X	X
11	X	X	X	
12		X		
13	X			X
14	X	X	X	X
15	X			
	9	9	5	8

As can be observed in Table 5 only fifteen respondents were interviewed and not twenty as originally intended, in addition very few of the respondents subscribed to a single designation which did not facilitate grouping as intended in Table 4. Findings not depicted here by way of a table or chart, are that all respondents confirmed their primary involvement in software estimation, development and implementation as well as exposure to bespoke development.

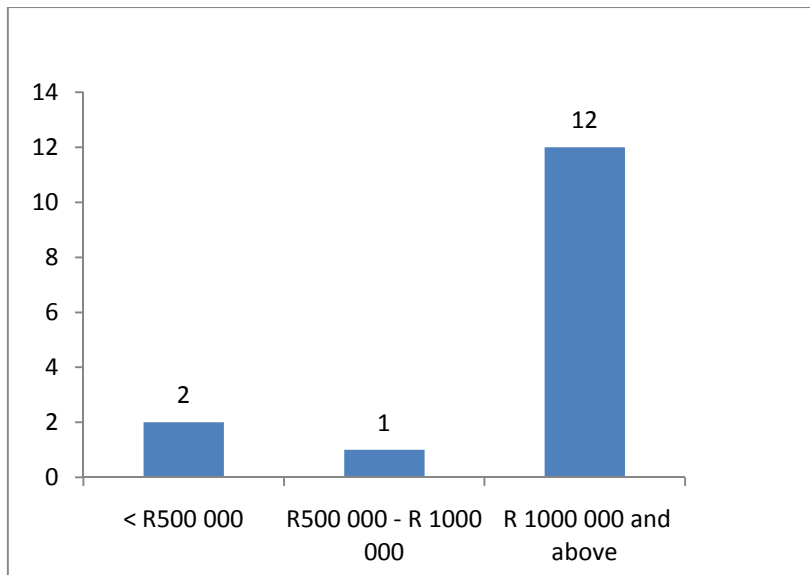


Figure 4: Project Experience by Respondent in Rand Value

Not all fifteen respondents had been involved or exposed to a software development project of one million rand and above although twelve of the fifteen or expressed as a percentage, 80% had.

4.3 Results pertaining to Proposition 1

Proposition 1: Expert estimation techniques are preferred to model-based techniques in South Africa

(Lewis 2001; Jorgensen 2004; Stellman and Greene 2005)

As a precursor to the identification or preference of techniques from the respondents, they were first asked in the questionnaire (Appendix A) to indicate whether or not they made use of an estimation technique and why. In response, all respondents made use of an estimation technique and the most common responses were that this was “part of company methodology” and that a technique provided “predictability”.

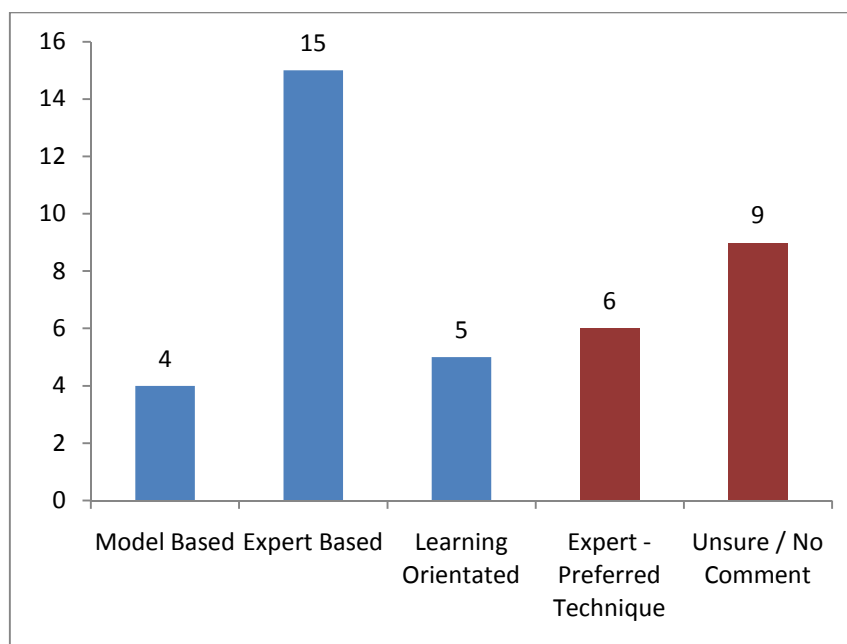


Figure 5: Named Techniques Employed By or Known to Respondents

The respondent's identified three named techniques that were presented to them from a list of six techniques. In Figure 5, model-based, expertise-based and learning orientated are the only techniques that were recognised or

identified by the respondents. Of the three techniques recognised, six believed that expertise-based estimation was the preferred method of estimation and the remaining nine respondents were unsure or unable to comment on a preferred method.

Table 7: Respondent Experience of the Use of Estimation Techniques in South Africa

Respondent	i	ii
1	Very Informal Approach	
2	No knowledge of the use of named/formal techniques	
3	Have not seen formal estimation techniques in practice	
4	Optimistic in general	Poor application & execution
5	Mainly a "Thumb Suck" Approach	
6	Informal/ Unstructured	
7	Unprofessional / Informal Approach	Customers not interested - Estimate must match their budget
8	Focus on quote to be awarded the business	Budget down approach
9	No real exposure to market place but not aware of any specific estimation techniques	
10	Largely a "Thumb suck" - not structured	
11	Very rarely a formal process i.e. Informal approach	
12	Informal - Unstructured - Thumb suck	
13	No/very little use of estimation techniques in SA	
14	Thumb suck Approach for vast majority	T&M arrangements that don't require an estimate i.e. open ended project
15	Very Informal Approach	

Table 6 illustrates the comment/s made by respondents regarding their experience of the use of estimation techniques, if at all, in the South African marketplace. All comments that are similarly worded or convey the same sentiment have been highlighted in the first column. The second column contains further comments made by the same respondents.

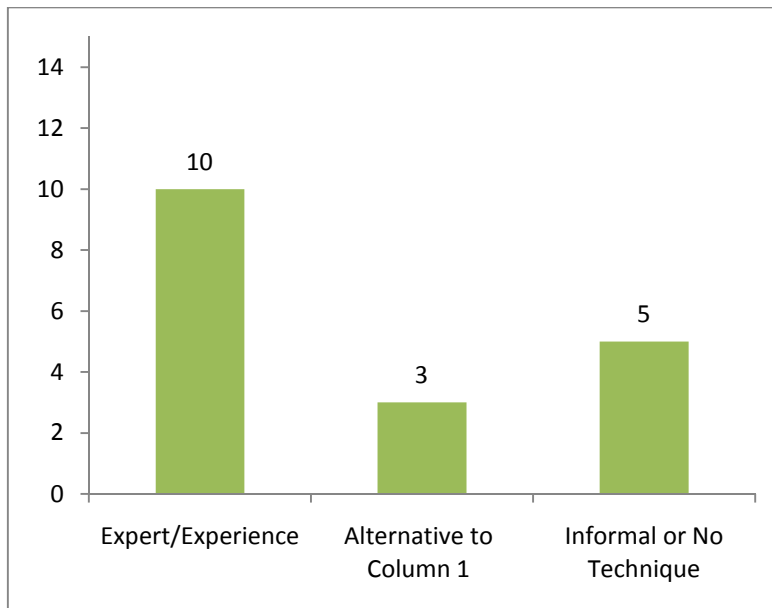


Figure 6: Estimation Technique prevalence in the South African Marketplace

Ten respondents indicated that of any known technique, experience or expertise-based estimation was most prevalent in South Africa, three respondents mentioned three alternative techniques as also being prevalent in South Africa. Five respondents indicated informal use of or absence of an estimation technique as displayed in Figure 6.

In addition to the findings in Figure 6 the respondents were asked in section D, question 4, of the questionnaire whether they agreed or disagreed with certain best practices advocated by estimation academics. With regards to the best practice of making use of good estimation experts familiar with the project's domain and background (Sheppard and Kadoda 2001; Jones 2005; Jorgensen 2007), twelve of the fifteen respondents agreed with this principle.

4.4 Results pertaining to Proposition 2

Proposition 2: Business interests as opposed to project scope are the greatest influence in arriving at software estimates

(Stellman and Greene 2005; Jorgensen 2007; Jones 2008; Jorgensen and Grimstad 2008)

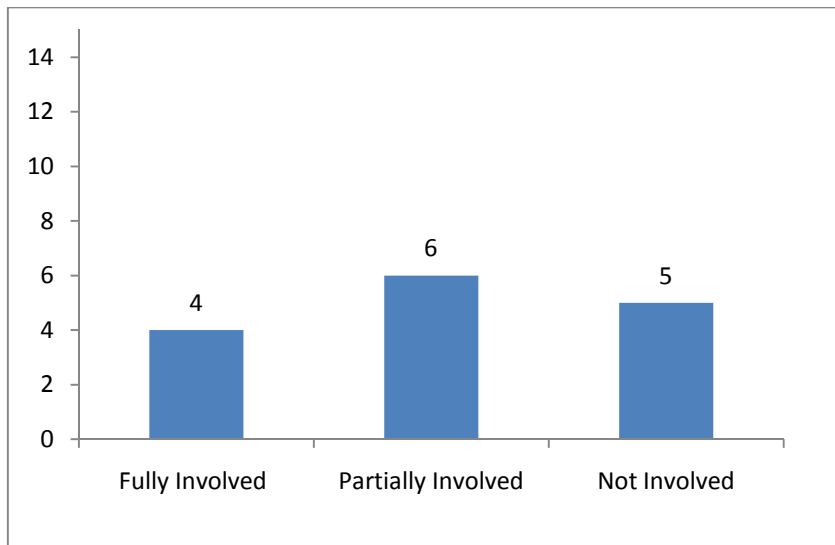


Figure 7: Level of Customer Involvement in the Estimation Process

Figure 7 depicts the respondent's views on the level of involvement the customer has during the estimation process. The responses range from fully involved to not involved. Partially involved is the most popular response albeit marginal in comparison to the other two scores.

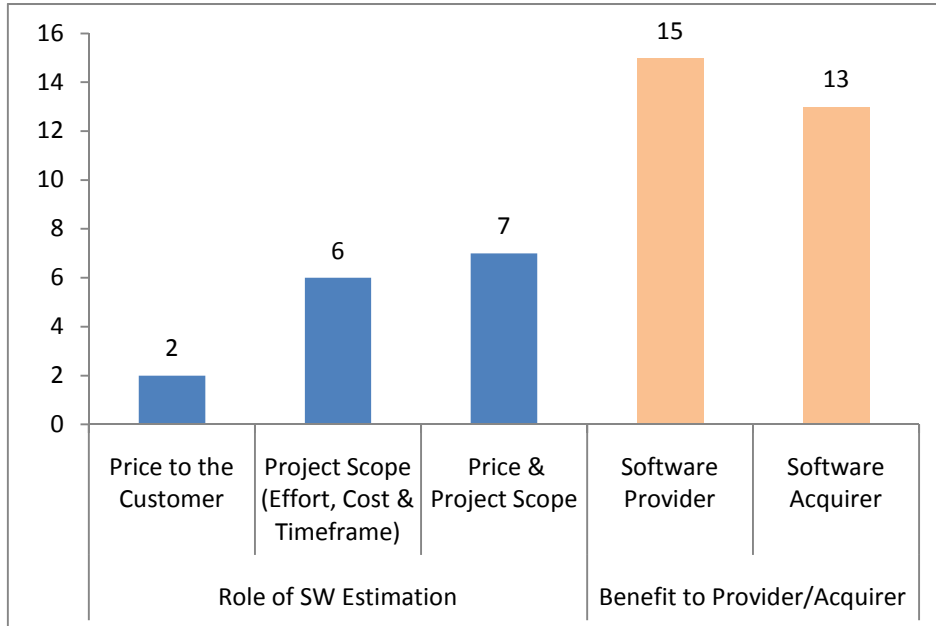


Figure 8: Role of Estimation Process and Benefit to Provider and Acquirer

In Figure 8 "Price to the Customer" and "Project Scope" were the two main roles that the estimation process fulfilled according to the respondents, however, the majority of responses were in favour of fulfilling both price and project scope.

Not illustrated in the figure but also recognised to a far lesser degree were the roles of creating a Work Breakdown Structure (4) and Identifying Risk (3). All the respondents felt that the software provider benefited from undergoing an estimation process while a very high percentage (86.6%), believed the software acquirer also benefited from the same process.

Table 8: Motivation for Software Estimates being too Low

Respondent	Business Interests	Project Scope
1		Functionality & Scope not fully understood
2		Not managing scope
3		Additional requirements brought in
4	Sales Pressure - make sales irrespective of cost	Oversight of the Schedule - Customer Side
5		Detail of Specifications - inadequate
6		Don't understand user/client requirements
7		Project not investigated properly; Commitment asked for too early with too little information available
8		No clear assumptions and ambiguity of project scope; Too little detail provided by the customer; Too much of a high level perspective taken
9		Insufficient knowledge of actual project - effort, resources etc.
10	Attempting to keep costs done; Meet price/budget without changing specifications	Scope and Specifications poorly defined; Poor interpretation of user requirements
11		Not understanding scope (under estimating); Failure to understand complexity of SW being developed; No clearly defined requirements
12	Tender Process - match price	
13	SW Provider desperate for work	
14	Simply meeting customer price/budget	
15	Sales influence to lower price & be awarded business	
	6	11

The respondents' responses to question 3 in the questionnaire are consolidated in Table 7 into the categories of Business Interests and Project Scope for the purpose of Proposition 2. The question allowed for multiple responses from each respondent and when totalled eleven responses were in favour of "Project Scope" and six were in favour "Business Interests". Two respondents answered that both categories were a reason for an estimate to be too low. The remaining

responses were not related to these areas however; there were other common reasons for estimates being too low which are listed below:

- Developers overestimating their abilities (10)
- Estimates not being done by experienced software practitioners (2)
- No formal estimation process used (2)

The respondents were asked the same question where the motivation was for estimates that were too high. There was far less response to this question however, for completeness, the following response was recorded amongst a few other unrelated comments:

- Estimators inflated or padded their estimates to mitigate the risk of underestimating (10)

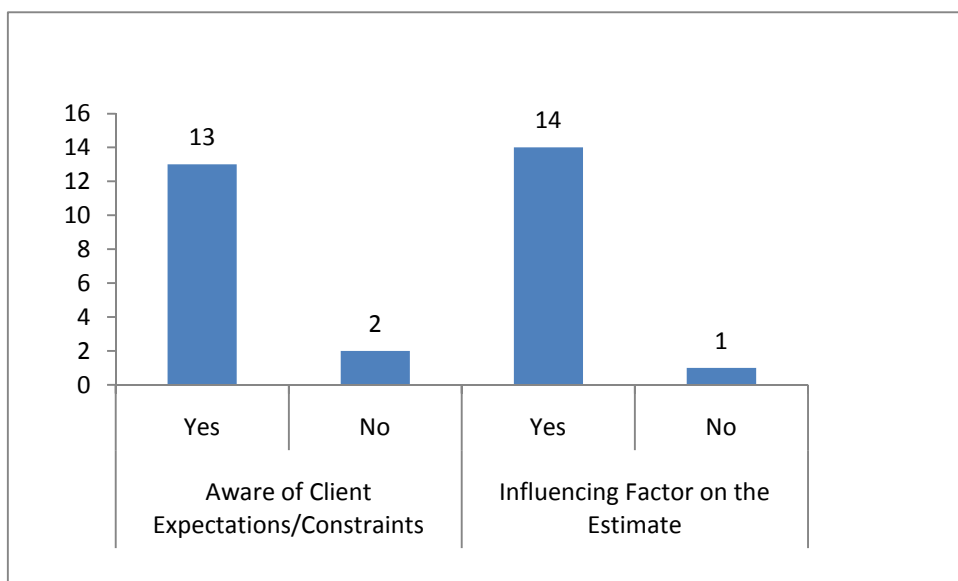


Figure 9: Clients Financial Needs/Constraints and the Influencing Factor

Leading from Table 8 is Figure 8 which represents respondents that are aware of the client's expectations or financial constraints when creating an estimate and whether they believe this awareness is an influencing factor on producing the estimate. In both instances almost all respondents are both aware of the financial expectations or financial constraints of the customer and believe that this is an influencing factor when producing the estimate.

Supplementing this chart are two prominent groups of responses pertaining to why the respondents believed that being aware of the client's expectations or financial constraints was an influencing factor. The first group of responses centres on adjusting the estimate to meet the budget, or phrased differently, to meet customer's financial expectations and be awarded the business. The second group of responses focuses on recognising the client's expectations or financial constraints and tailoring the scope to match the budget or expectation also with the intention of being awarded the business.

The final results to be discussed under Proposition 2 relate again to the best practices referred to in section D question 4, there are two best practices that are applicable here. Firstly that estimators must avoid conflicting goals of price/cost versus project effort required (Jorgensen and Grimstad 2008) and secondly that estimators must be able to justify and criticise their estimates and remove pressures exerted by the different stakeholders (Jones 2005). The response for the first best practice was that twelve out of fifteen respondents agreed and fourteen agreed with the second best practice.

4.5 Results Pertaining to Proposition 3

Proposition 3: Estimate values are committed to at the start of the project and are seldom revised

(Ross 2003; McConnell 2006)

Table 9: Revision of the Software Estimation during the Project's Progress

Respondent	Yes	No	Project Progress
1		X	
2	X		40%
3	X		Ongoing
4	X		20% and 40%
5		X	
6	X		40%
7	X		No specific point
8	X		Frequently based on timeframes
9	X		60%
10	X		60-80%
11	X		No specific point
12	X		20 & 40%
13	X		20-30%
14	X		20-30%
15	X		40%
	13	2	

Table 8 illustrates which respondents believed a software estimate should be revised after the initial presentation and at what point in the projects progress as opposed those that did not believe the initial estimate should be revised. Of those respondents that would revise the estimate the most common point of the revision was be between 20% and 40%.

In section C question 1 of the questionnaire the respondents were given six reasons, taken from the literature, as to what they believed the greatest difficulties were in creating an accurate software estimate. The respondents were able to choose more than one reason, of the six provided, the top three reasons were:

1. Completeness of requirements (Functional, Non functional and technical) – 13 responses
2. Accuracy of input variables – (Software Size, complexity and technical requirements) – 12 responses
3. Level of Uncertainty – 10 responses

No other reason scored more than 8 responses.

With regards to the best practices presented to the respondents nine agreed with the concept of assessing the level of uncertainty of the estimate and monitoring changes over time. Further to this the best practice proposes that the estimate is presented as probability distribution rather than a single value (Ross 2003; Dekkers 2005; McConnell 2006).

4.6 Results Pertaining to Proposition 4

Proposition 4: Despite a combination of estimation techniques being advocated by academics this is seldom practised in reality

(Boehm, Abts and Chulani 2000; Jorgensen 2004; Jones 2005)

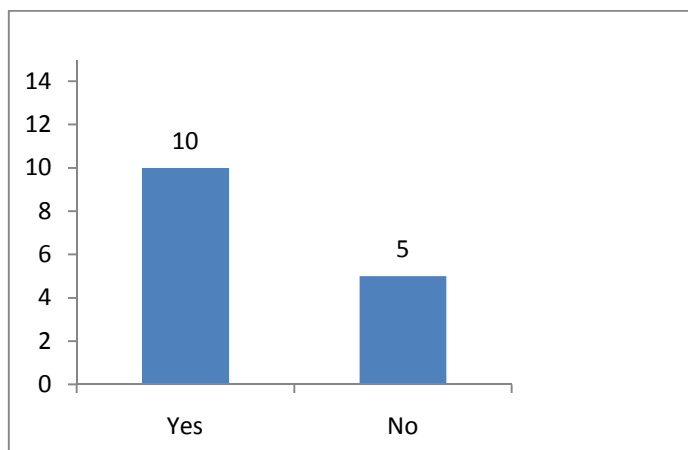


Figure 10: Respondents Familiar with the Combining of Estimation Techniques

66% of the respondents indicated that they were familiar with the concept of combining estimation techniques when doing a software estimate as per Figure 10.

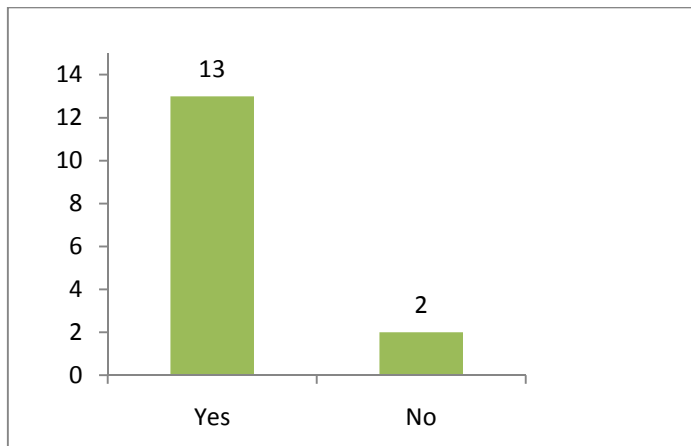


Figure 11: Respondents that agreed with the statement that the Combination of Estimation Techniques is advocated by Academics but seldom practised in Reality

Figure 11 indicates that almost all of the respondents felt that estimation techniques were seldom combined during the software estimation process despite being advocated by academics.

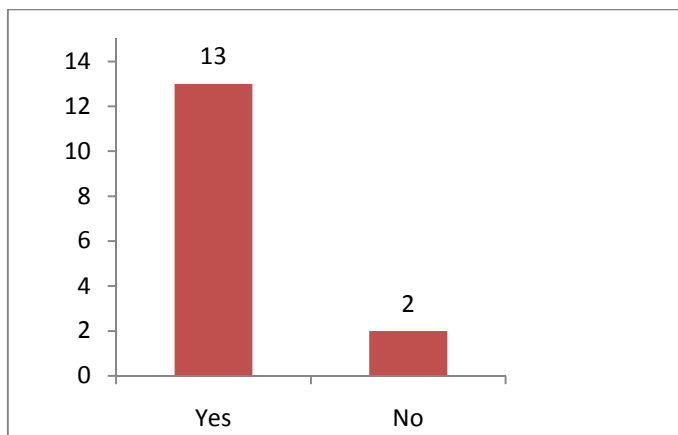


Figure 12: Respondents that believed that the Combination of Estimation Techniques provides more reliable Estimates

Similar to Figure 11, Figure 12 reflects that the majority of respondents believed that combining estimation techniques would provide a more reliable software estimate. The respondents that disagreed in Figure 11 and 12 were not the same respondents.

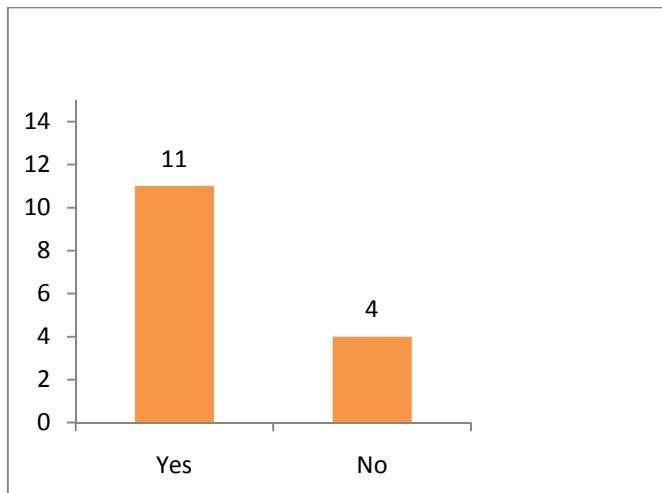


Figure 13: Respondents that believe it is Possible to Combine Estimation Techniques in Reality

Similarly worded to Figure 11 but different in meaning, Figure 13 shows those respondents that believed it would be possible to combine estimation techniques when providing a software estimate. No constraints were specified in this question and of the reasons given as to yes or no, no pattern or real similarities were amongst the reasons however, for interest purposes, reasons such as “time consuming”, “expertise available” and “the level of investment required by the software provider up front” were mentioned for the negative responses.

The respondents were also asked to comment on the best practice of combining estimates from different experts and estimation strategies as well as assessing the differences in their results (Boehm, Abts and Chulani 2000; Jones 2005). Ten of the respondents agreed with this best practice.

4.7 Summary of the results

4.7.1 Demographic Profile of Respondents

Only 15 respondents target were interviewed as opposed to the target of 20, of these, most respondents had fulfilled many of the roles named in Table 4. As a result it was not possible to group the sample’s responses by respondent type. 80% of the respondents had been, or were involved in a software project that

valued one million Rand or more to the customer. All the respondents had bespoke software development exposure and software estimation experience.

4.7.2 Proposition 1 Summary

Proposition 1: Expert estimation techniques are preferred to model-based techniques in South Africa

(Lewis 2001; Jorgensen 2004; Stellman and Greene 2005)

All respondents indicated that they made use of a software estimation technique. Three 'named' techniques were identified by the respondents, (1) Model Based, (2) Expert Based and (3) Learning Orientated. Six of the respondents perception was that expert based was the preferred technique, the remaining respondents were unsure or could not confidently comment.

When commenting on their experience of the use of estimation techniques in South Africa, a large majority of the respondents had very similar comments, namely, "an informal approach"; "no evidence of the use of formal techniques" and "unstructured" or "thumb suck". From a South African perspective ten respondents indicated that experience or expert-based software estimation was the most prevalent technique, five respondents, however, believed that an informal or no technique was most prevalent.

Twelve of the fifteen respondents agreed with the best practice of making use of good estimation experts, familiar with the projects domain and background (Sheppard and Kadoda 2001; Jones 2005; Jorgensen 2007).

4.7.3 Proposition 2 Summary

Proposition 2: Business interests as opposed to project scope are the greatest influence in arriving at software estimates

(Stellman and Greene 2005; Jorgensen 2007; Jones 2008; Jorgensen and Grimstad 2008)

Ten of the fifteen respondents said that customers were either fully involved or partially involved in the estimation process, the remainder said that the customer was not involved. The 'Price to the Customer' and 'Project Scope' was identified as the two main roles fulfilled by software estimation according to the respondents. The majority felt that both roles were fulfilled as opposed to a single role in isolation, although project scope far outweighed price to customer.

All the respondents felt that the software provider benefited from the estimation process and 86% believed that the software acquirer also benefited from the process. When assessing estimates that were too low, eleven responses favoured project scope and six business interests or price (respondents were not restricted to a single answer). Other reasons were provided for estimates being too low but they were not related to project scope or business interests. The only common response for estimates being too high was due to estimators inflating their estimates in an effort to mitigate risk.

At least thirteen of the respondents were aware of the client's financial expectations or constraints and believed this to be an influencing factor when creating the estimate. Two rationales were provided as to why this was an influencing factor, the first resulted in the estimate being adjusted to match or meet the client's expectations and be awarded the business. The second was to tailor the projects scope in order to meet the client's expectations and therefore be awarded the business.

Two best practices were relevant to proposition 2, one which states that estimators must avoid conflicting goals of price/cost vs. Project effort (Jorgensen and Grimstad 2008) and the other that estimators must be able to justify and criticise their estimates and remove pressures exerted by different stakeholders (Jones 2005). Twelve respondents agreed with the first best practice and fourteen with the second.

4.7.4 Proposition 3 Summary

Proposition 3: Estimate values are committed to at the start of the project and are seldom revised

(Ross 2003; McConnell 2006)

Thirteen respondents answered that the initial estimate should be revised as the project progresses over time, the most common point in time was between 20 to 40% of the projects progress.

Of the top three reasons given by the respondents regarding the greatest difficulties in creating an accurate software estimate, the 'level of uncertainty' was one of these reasons. Nine respondents agreed with the concept of assessing the level of uncertainty of the estimate and monitoring changes over time (Ross 2003; Dekkers 2005; McConnell 2006). This best practice proposes that the estimate is presented as a probability distribution rather than a single value.

4.7.5 Proposition 4 Summary

Proposition 4: Despite a combination of estimation techniques being advocated by academics this is seldom practised in reality

(Boehm, Abts and Chulani 2000; Jorgensen 2004; Jones 2005)

Ten respondents were familiar with the concept of combining estimation techniques. Thirteen respondents felt that estimation techniques were seldom combined when providing an estimate, despite this being advocated by academics. Thirteen of the respondents also believed that combining estimation techniques would provide a more reliable estimate. Fewer respondents, eleven, believed that it was possible to combine estimation techniques in reality when constructing a software estimate. Ten of the sample agreed with the best practice of combining estimates from different experts and estimation strategies and assessing the differences in their results (Boehm, Abts and Chulani 2000; Jones 2005).

CHAPTER 5: DISCUSSION OF THE RESULTS

5.1 Introduction

This chapter expands on Chapter 4 in that it provides for the discussion of the results and comparison with the information presented initially in Chapter 2 being the literature review. Following the same format as the previous chapter it starts with a discussion around the demographic profile of the respondents followed by discussion and comparison of Proposition 1 through to Proposition 4. A conclusion of each proposition is presented as a sub-heading of each proposition providing a culmination of the findings and the literature. The chapter concludes with a summary of the entire chapter and a restatement of the conclusions of each proposition.

5.2 Demographic profile of respondents

The population from which the sample in Table 4, Chapter 4, originated had three parameters outlined in Chapter 3; firstly, they had to operate primarily within the realm of software estimation, development and implementation, secondly, the population had to have undertaken a software project in the last two years or were in the process of undertaking a software project, lastly the project was ideally no less than one million rand in value. Overriding these parameters is the focus of the thesis around bespoke software development as bulleted under the delimitations of the thesis in Chapter 1.

As can be observed in Table 5, Chapter 4, only fifteen respondents were interviewed and not twenty as originally intended. In addition very few of the respondents subscribed to a single designation which did not facilitate grouping as intended in Table 4. The primary reason for the actual outcome of the sample was that the skill set and experienced needed to respond to the questionnaire was relatively rare. The intention to obtain estimation feedback from specific groups of software practitioners was mistaken in the fact that it appears that only multi skilled software practitioners have the ability to respond with a high level of confidence. The result was that the organisations

encountered by the author had one and occasionally two estimation representatives available. Referrals by the representatives of these organisations, while helpful, often referred back to organisations already interviewed indicating a limited pool within the network the author attempted to exploit.

Not all fifteen respondents had been involved or exposed to a software development project of one million rand and above although twelve of the fifteen or expressed as a percentage, 80% had. This outcome appeared to be simply a result of the project size that some of the smaller organisations interviewed were limited to. Common to the objective of the research methodology and the results obtained, all the respondents confirmed their primary involvement in software estimation, development and implementation as well as exposure to bespoke development.

5.3 Discussion pertaining to Proposition 1

Proposition 1: Expert estimation techniques are preferred to model-based techniques in South Africa

(Lewis 2001; Jorgensen 2004; Stellman and Greene 2005)

In the literature review, the research discussed identifies project management as empirically linked to successful software projects according to Gowan and Mathieu (2005). Within Software Project Management (SPM) Software Effort & Cost Estimation is listed as one of the key twelve components of SPM by Nienaber and Cloete (2003). Further on in the review under the discussion of 'Estimation Techniques' are the different methods of software estimation discussed, namely expert estimation and model-based estimation.

Jones (2007) differentiates expert from model estimation only in the final quantification step of the estimate, if this step is judgement based then the technique is expert based and if mechanical then model based. Jorgensen (2004) described expert estimation as a method that exists between unstructured intuition and expert judgement, the latter based on historical data,

procedures and structure. Jorgensen (2004) also claims expert estimation to be the most prominent technique. Considering this description and the research findings, the following points are deemed most relevant in support of Proposition 1:

- Of the six named techniques presented to the respondents, Expert estimation was the only method recognised by all respondents
- Six respondents felt that from a broad perspective expert estimation was the preferred method of estimation, the remaining respondents were unable to comment
- When commenting on their experience of the use of estimation techniques in South Africa, the majority of the sentiment was a mixture between little or no evidence of formal techniques and guesswork
- When looking at which estimation techniques were most prevalent in South Africa, ten of the fifteen respondents first response was that expert or experience based estimation was most prevalent in South Africa
- Considering the finding immediately above, the remaining five respondents felt that no technique or an informal approach was most prevalent

In an attempt to find some insight as to why expert estimation is preferred to model based estimation, Table 2 in the literature review contains two relevant points. Firstly, one disadvantage of model based techniques is that few organisations fully understand the workings of the models and are not comfortable using them. Secondly, an advantage of expert based techniques is that they are useful in the absence of empirical data, experts are deemed to have a natural advantage in these instances (Boehm et Al 2000 and Jorgensen 2004, 2007). What is relevant considering the second point and not addressed in the literature, is how many projects do or do not have empirical data available.

There is a final key point regarding the two different estimation techniques discussed in the literature review and not apparent or forthcoming from the findings in Chapter 4. That is that the accuracy of the estimate is paramount and much of the debate surrounds the issue of which technique will provide the most accurate or reliable estimate. Jones (2005) found automated or model based estimates to be more accurate in larger projects and that the most successful projects made use of model base techniques. Stellman and Greene (2005) believed expert estimation to be the most precise method in those projects they encountered. While twelve of the fifteen respondents agreed with the best practice of making use of a good estimation expert with the applicable domain background experience, this practice speaks mainly to using the relevant domain background to correctly calibrate or apply the respective estimation technique in an effort to produce the most accurate estimate and may have been misinterpreted by the respondents.

5.3.1 Conclusion – Proposition 1

The conclusion is that when testing Proposition 1 in the South African context the findings are that expert estimation techniques are preferred to model-based techniques. Secondly, the findings oscillate mainly between expert-based estimation and an informal or “no structured approach” to estimation. One of the indications of this could be that, at best, software practitioners in South Africa make use of expert or experienced based estimation and the remainder of estimation is done on an informal basis. It is evident however, that based on the findings presented up until this point that knowledge and experience regarding software estimation or the different methods of estimation is lacking in general. The findings of a more technologically mature country may have more depth and diversity regarding the knowledge and application of other estimation techniques.

5.4 Discussion pertaining to Proposition 2

Proposition 2: Business interests as opposed to project scope are the greatest influence in arriving at software estimates

(Stellman and Greene 2005; Jorgensen 2007; Jones 2008; Jorgensen and Grimstad 2008)

McConnell (2006) and David (2006) define project scope as incorporating software size and the size of the project, typically measured in effort, cost, features or functionality and the timeframe of the project. Leading from this is the role of software estimation which attempts to accurately define project scope. Cost within the definition is specific to the cost of delivering the project by the software provider. The role of software estimation as far as the respondents were concerned was less in agreement with the academic definition (s). The main difference of opinion was not that the role of the estimate was to establish the scope of a project but that it was also intended to establish the price of the project for the customer.

Two thirds of the respondents indicated that the customer was fully or partially involved in the estimation process, the assumption is that this involvement centred on defining the project scope although it is not unreasonable to assume that the customer's financial expectations were imparted during this process. Despite the latter being assumed it is evident in both the findings and literature review that the belief is that the estimation process benefits both the software provider and the acquirer (Boehm, Abts and Chulani 2000, Jorgensen 2007). Jones (2008) however, reported the following on a sample of software litigation cases:

- The primary cause of the litigation was due to the absence of an accurate estimate or the rejection of the initial estimate
- 50% of the cases had a competent estimate which was replaced to suite the organisations needs and not the project teams capabilities

The findings in this instance support the literature, thirteen respondents indicated that they were aware of the client's expectations or constraints and

fourteen of the fifteen believed that this knowledge was an influencing factor when creating the estimate. The respondents also provided two likely responses by the software provider; the estimate would be matched with the client's financial expectations or constraints, alternatively the project scope would be reduced to accommodate the client's budget. Both responses have the objective of being awarded the business.

When looking at software estimates and the influences that cause them to be either too high or low, Jorgensen (2007) found that over optimistic or low estimates were provided in an attempt to serve business interests or as a result of overconfidence on the software provider's part. Stellman and Greene (2005) found that estimates were too low due to provider overconfidence, an aggressive stance by management or unrealistic goals. Interestingly the respondents provided eleven project scope related reasons for an estimate being too low as opposed to only six business related reasons. The respondents and the literature alike stated that estimates that were too high were as a result of 'padding' to create additional time and reduce the risk of the project.

In order to avoid estimates that are either too high or too low, Jorgensen and Grimstad (2008) advocate that information not relevant to the fulfilment of the project, like a customer's cost expectations, should be filtered out irrespective of the estimation technique used. This sentiment is carried forward in one of the relevant best practices by the same authors which states that estimators should avoid conflicting estimation goals. Jones (2005) says that estimators should justify and criticise their estimates, avoid pressure from clients and management alike to provide over optimistic estimates. Twelve respondents agreed with Jorgensen and Grimstad and fourteen with Jones.

5.4.1 Conclusion – Proposition2

Where we look at the role of the software estimate we find that although South African software practitioners only agree in part with the literature, they also believe that the estimate is intended to provide a price to the customer. The majority of the respondents experience was that the customer was involved, to

varying degrees, in the estimation process. A high percentage of the respondents were also aware of the customer's financial expectations or constraints and an even higher percentage believed that this was an influencing factor in the creation of the estimate.

When identifying the reasons for estimates being too low, as this appears most common, the academics cited a mixture of reasons relating to business interests and overconfident software providers. The respondents cited similar business reasons but were more in favour of the underestimation of project scope as the main cause. The best practices advocated by the academics appear to specifically appeal to the dilemma of business interests versus project scope. Both practices encourage estimators to concentrate on the projects scope and avoid information not relevant to its fulfilment, also to resist stakeholder pressures which lead to the estimate being over optimistic.

The conclusion is that despite a strong belief amongst software practitioners that project scope is the primary driver behind the formulation of a software estimate, ultimately business interests prevail and the estimate is then aligned with the business interests of the provider and the acquirer. This judgment is underpinned by the strong correlation between the litigation findings of Jones (2008) and the two scenarios provided by the respondents when aligning the estimate with the customer's financial expectations or constraints.

5.5 Discussion pertaining to Proposition 3

Proposition 3: Estimate values are committed to at the start of the project and are seldom revised

(Ross 2003; McConnell 2006)

Ross (2003) states that some of the most critical decisions pertaining to a project are made when the least is known about the project, in other words the start of the project. It is at this point that that uncertainty is at its greatest according to Ross. McConnell (2006) uses the "Cone of Uncertainty" to illustrate a similar point, at the start of the project the cone is at its largest yet it is at this

point that software companies make undertakings too early which weakens the predictability of the estimate, increases its risk and compromises the success of the project.

In correlation with the literature the respondents identified uncertainty as one of the top three most difficult aspects involved in the creation of an estimate. Nine of the fifteen respondents also agreed with the best practice of assessing the uncertainty of the project and monitoring the changes over time (Ross 2003). Furthermore the best practice advocates that customers should be advised that the estimate cannot remain fixed and therefore should be represented as a probability distribution due to the high variability of the project at its beginning (McConnell 2006). Thirteen respondents agreed with revising the estimate after its initial presentation or start of the project and identified the most common point of revision of the estimate to between 20 and 40%. This finding maps closely to McConnell's "User Interface Design Complete" stage shown in Figure 14 which is roughly between 35-40% on the timeline and where the "Estimate Variability" is drastically reduced. A smaller majority agreed with presenting the estimate as a probability distribution rather than a single value.

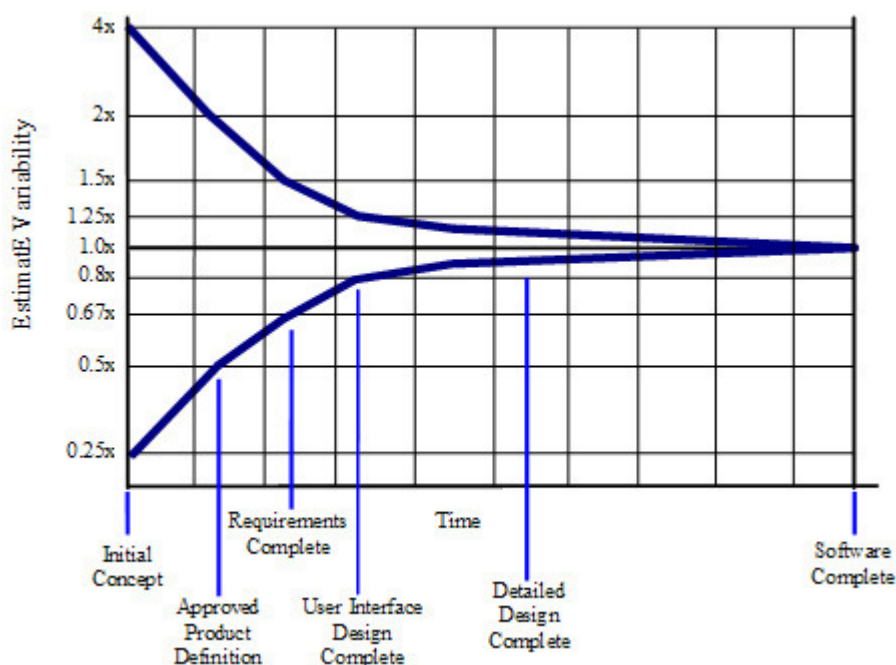


Figure 14: The Cone of Uncertainty (McConnell 2006)

5.5.1 Conclusion – Proposition 3

The conclusion is that estimation values are in fact revised after the start of the project and seldom remain unchanged after the project commences. Not all respondents agree as to when the estimate should be revised but there is a common band between 20 and 40% which concurs with the literature, the respondents are also somewhat divided as to whether the estimate should be presented as a single value as opposed to a probability distribution. However, it does appear that the majority of software providers in South Africa follow a method of revising the estimate values after the initiation of the project which aligns closely with the findings in the literature.

5.6 Discussion pertaining to Proposition 4

Proposition 4: Despite a combination of estimation techniques being advocated by academics this is seldom practised in reality

(Boehm, Abts and Chulani 2000; Jorgensen 2004; Jones 2005)

Table 2 in the literature review compares the advantages and disadvantages of model vs. expert estimation as per Boehm et Al. (2000) and Jorgensen (2004, 2007). Using this comparison as an example it appears that certain techniques are better suited to certain projects, from this, it would seem that the next logical step would be to combine estimation techniques in an effort to guarantee a more reliable estimate.

Jorgensen (2004), based on his findings, proposes the best practice of “Combing estimates from different experts and estimation strategies”. This best practice is supported and recommended by Boehm, Abts and Chulani (2000) and Jones (2005). Specifically Boehm et Al, refer to the combining of techniques as a ‘composite’ method, the point being not favour one technique but to determine why one estimate produces a marked difference from the other. Ten of the respondents agreed with this best practice however, only ten of the respondents were familiar with the practice of combining estimation techniques.

The following findings were also deemed pertinent to this proposition:

- Thirteen of the fifteen respondents agreed with the statement that the combination of estimation techniques was advocated by academics but seldom practiced in reality
- Thirteen of the respondents also agreed that the combination of estimation techniques would provide a more reliable estimate
- Fewer respondents, eleven, believed that it was possible to combine estimation techniques in reality

5.6.1 Conclusion – Proposition 4

The majority, or no less than ten respondents, indicated that they were aware of the practise of combining estimation techniques and agreed that this would logically provide a more reliable estimate; they also maintained that combining estimation techniques in reality would be possible. However, as an outright finding in Figure 11 of Chapter 4, thirteen of the respondents supported the final proposition that despite a combination of estimation techniques being advocated by academics this was seldom practiced in reality.

In an attempt to understand this, the anecdotal comments of the negative responses pertaining to why it would *not* be possible to combine estimation techniques in reality are considered. The comments, “time consuming”, “expertise available” and “the level of investment required by the software provider up front” may be factors that explain why many software providers are not subscribing to the proposed best practice. More substantively the overall approach to estimation and the preference towards expert based techniques, if any, explored under Proposition 1 in South Africa could indicate that the combining of estimation techniques may be an evolutionary step that has not yet been taken.

5.7 Conclusion

The demographic profile of the respondents is discussed at the beginning of the chapter, it was established that only fifteen of the twenty respondents were interviewed and that the intended software professional groups did not exist in their proposed form. In the writers opinion this did not detract from the credibility of the information obtained but did affect the nature of the reporting of the findings. Furthermore the remaining of the objectives and parameters set out in the Research Methodology chapter were met as far as the demographics of the respondents were concerned.

Proposition 1 follows the opening discussion of the respondents and concludes that with regards to “expert estimation techniques being preferred to model based techniques in South Africa” that expert estimation is preferred to model based estimation. In addition it is added that the extent of estimation methods in South Africa vary between expert-based and an unstructured or informal approach.

Proposition 2 deals with whether “Business interests as opposed to project scope are the greatest influence in arriving at a software estimate”. In this regard the conclusion was that although there was a strong belief amongst the respondents that project scope was the primary driver behind the formulation of a software estimate, ultimately business interests prevailed and the estimate was aligned with the business interests of the provider and the acquirer. The literature presented in this regard indicated a similar sentiment.

Proposition 3 states that “Estimate values are committed to at the start of the project and are seldom revised”. Despite the fact that although not all respondents agreed as to when an estimate should be revised or if it should be presented as a probability distribution or single value, contrary to the proposition, estimate values are revised subsequent to the inception of the project and in a timeframe that corresponds favourably with the literature put forward.

Proposition 4 questions if “Despite a combination of estimation techniques being advocated by academics this is seldom practiced in reality”? The conclusion was in support of the proposition, even though the majority of respondents subscribed to the merits of the proposition as a best practice put forward by academics, they did not believe that this was a reality at present.

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

The final chapter brings together the conclusions of the study, both the literature and the findings are discussed in line with the context of the entire study outlined in Chapter 1. Following the conclusions of the study are the recommendations by the author to the stakeholders initially identified, in light of the outcomes of the research undertaken. The chapter closes with suggestions for further research involving areas that the author feels may be of interest and relevance to further study.

6.2 Conclusions of the study

6.2.1 Proposition 1: Expert estimation techniques are preferred to model-based techniques in South Africa

(Lewis 2001; Jorgensen 2004; Stellman and Greene 2005)

The respondents were only able to identify three named techniques, based on their responses, expert estimation was identified as the preferred technique. As far as the practice of software estimation in South Africa is concerned, expert or experience based estimation is the most prevalent, with a strong perception that much of the software estimation in South Africa is done largely an informal or unstructured process. The findings are therefore in favour of Proposition 1.

Much reference has been made regarding the importance of software estimation within a software development project and its role in trying to ensure the project is successful by being on time and within budget. One of the main purposes of this study was to gain insight into the different estimation approaches applied in the country. Considering this insight in conjunction with the literature on estimation techniques, it would appear that the overall knowledge and skills surrounding estimation in South Africa is lacking or

perhaps still maturing by comparison to first world countries such as the U.S., Europe and Japan. In order for software estimation to improve as a process in South Africa, better education of the estimation methodologies and the application of more formal and diversified methodologies needs to come about.

6.2.2 Proposition 2: Business interests as opposed to project scope are the greatest influence in arriving at software estimates

(Stellman and Greene 2005; Jorgensen 2007; Jones 2008; Jorgensen and Grimstad 2008)

The findings reveal that the respondents believed that the role of software estimation was to define the project scope as well as a price to the customer, scope outweighing price in terms of importance in this instance. When assessing estimates prepared for projects that were too low, reasons relating to project scope once again outweighed business interests. The respondents added that they were often aware of the client's financial expectations and that this influenced the estimate. Software providers according to the responses are prepared to compromise the price in order to be awarded the business, or, more logically, tailor the scope of the project to meet the client's expectations. Despite the academics cautioning against compromising the estimate for business reasons and the respondents agreeing with this principle, the influence of the clients financial expectations remains a very real factor and contributor to failed projects as cited in the literature by Jones (2008).

The findings pertaining to Proposition 2 are in contradiction to the proposition statement, however, it is the point at which the estimate iteration is aligned with the client's expectations and/or the software provider's ambition to be awarded the business that the findings in fact support the proposition. Based on the literature this appears to be an international trend which contradicts the purpose and objectives of the estimation process. Addressing this in order to improve the estimation process in South Africa is a complex problem as software providers need to secure business in order to remain operational.

6.2.3 Proposition 3: Estimate values are committed to at the start of the project and are seldom revised

(Ross 2003; McConnell 2006)

This proposition was not upheld by the findings, there were differences between the respondents although the majority indicated that software estimates are revised after the start of a project and the most common point for revision was between 20-40% of the projects timeframe. The rationale for revising the estimate in the findings concurred with the literature and was primarily based on the uncertainty that exists at the start of a project. These findings are favourable and in line with the best practices put forward by academia, in this instance this is positive and bodes well for the future of software development success in South Africa.

6.2.4 Proposition 4: Despite a combination of estimation techniques being advocated by academics this is seldom practised in reality

(Boehm, Abts and Chulani 2000; Jorgensen 2004; Jones 2005)

Although not all respondents were initially aware of the statement made in the final proposition, the concept and logic of combining estimation techniques was agreeable to the majority of the respondents'. Despite this the majority of the respondents sanctioned Proposition 4 and did not believe the practice of combining estimation techniques was taking place in reality. The proposition arises from a best practice put forward by some academics as a solution to overcoming the shortfalls inherent in single methodologies. Arguably the combining of techniques is onerous but it does appear credible solution as is seen in the findings mentioned above.

Taking into account the final comments made regarding Proposition 1, in conjunction with those of Proposition 4, it would appear that whether a viable solution or not, South Africa has some preliminary steps to take before a more advanced approach to software estimation can be taken. When looking at the

four propositions tested the findings were in support of three of the propositions. This would indicate that of the perceptions recorded pertaining to software estimation; the guiding principles and lessons learned internationally still plague the South African software development marketplace.

Given the key positioning of software estimation in the software development cycle it is imperative that the software providers in South Africa look to the international methodologies and best practices as a source of guidance towards formalising the estimation process. Where we consider the project size or value focused on within this research, it does not seem wise for the providers to be assuming an informal or unstructured estimation approach at such an early point of the project. South African software providers need to refresh their approaches to software development and produce sound, reliable software estimates which in turn will provide justification of cost to the customer and avoid risky compromises being made. Furthermore they should continue to do what is in line with best practice and constantly revise their estimates to manage the uncertainty of the project. Finally as part of the education and learning required around estimation techniques start to apply these techniques and where possible use combine them in order to establish a knowledge base of how to achieve successful software development projects.

6.3 Recommendations

The stakeholders identified in Chapter 1 were broadly the software vendor and the software acquirer or customer. Within those two groups of stakeholders, software developers and architects were named on behalf of the vendor and the IT Manager and CIO were named on behalf of the customer. Starting with the vendor, the software estimation process whether experience or model based should be a structured, formalised process. The intention of this is to avoid guesswork and the exclusion of important information relevant to the project. If no established estimation technique or methodology is known then the priority is to close the theoretical knowledge gap and combine that with the vendor's experience. Where possible apply the applicable methodology to the project that needs to be estimated if the skills are available, more so, apply more than

one estimation approach to a project if the investment in time and resources are justified by the project value and size.

The software developers and architects must produce an estimate that addresses the scope of the project and not the customer's expectations or constraints. If this is not possible then the scope and the functionality of the software must scale with the cost of the estimate and ultimately the price to the customer. The risk of conceding to the customers financial expectations without aligning the effort and cost involved will threaten the vendor's existence in the long term. In association with the software vendor being able to justify the estimate and price to the customer is the process of ongoing negotiation with the customer over the timeframe of the project. This involves the flexibility to revise the estimate after the project has commenced and re-negotiate the projects parameters, within reason, with the customer. While this practice is often difficult to achieve with the customer, educating the client around the software development process may assist with the understanding behind this.

The recommendations to the software acquirer or customer are to try and understand the software development process to the extent where it will impact on them. It is important for the customer to understand the development process in order to understand why changes to the original estimate may come about or even why their financial expectations are not in line with the scope required to deliver the software functionality expected. Flexibility around the project must be allowed for by the customer, once again within reason, larger projects require longer time and this often increases the level of uncertainty of the project making it more difficult to estimate outright at the start. No benefit is obtained if the customer attempts to commit the software provider to an initial price, timeframe or standard of quality up front, if the parameters of the project contain uncertainty at the beginning of the project. Inflexibility may result in the vendor withdrawing, the customer appointing a new vendor with similar results, the software itself compromised in one or more of the three parameters mentioned above. All of these factors greatly reduce the success of the project, the objective of the intended software for the business and its adoption into the existing systems. A positive relationship with the vendor, an insight into the

development process, involvement and support from the customer are all contributing factors to a successful software development project.

6.4 Suggestions for further research

In the author's opinion there is a great deal of scope in looking at the existence of any theoretical sources or education of software estimation in the country. It would appear that it is not dealt with at a tertiary level based on comments made by some respondents. Linked to this is what resources software development companies make use of in South Africa apart from the know-how of the skills employed by them. In the interest of improving this key aspect of software development it would be worth knowing if the extent of the software estimation expertise exists only within the software professional's community or whether multinationals, for example, in South Africa have a knowledge source that contains ideologies and methodologies that can be referred to.

Proposition 4 is also an area that can be expanded on, at the time of gathering all the literature there was no evidence of this best practice having been applied extensively and therefore inconclusive as a solution. There was also an absence of literature specific to South Africa regarding software estimation and any empirical based studies on the success rates of software development projects in the country and what the reasons behind those figures were, a study similar to the Standish Report could be replicated in South Africa.

Software estimation is potentially far more complex when looking at bespoke development, is the estimation of bespoke software development projects still trying to be perfected or has it been decided that the nature of bespoke projects needs to change in an attempt to simplify their complexity?

From a broader perspective, Javed, Manzil-E-Maqsood and Durrani (2006), made reference in their research relating to 'Geographically Distributed' projects, to cultural obstacles that prevented formal processes and techniques being carried out and conformed with. There may well be cultural aspects not evaluated in this research that dictate how and why South Africans have this approach to software estimation.

REFERENCES

Abdel-Hamid, T.K. and Madnick, S.E. (1989) Lessons Learned from Modelling the Dynamics of Software Development, *Management of Computing*, 32(12), pp. 1426-1455.

Aladwani, A.M. (2002) IT Project Uncertainty, Planning and Success - An empirical investigation from Kuwait, *Information Technology & People*, 15(3), pp. 210-226.

Armour, P.G. (2004) Real Work, Necessary Friction, Optional Chaos, *Communications of the ACM*, 47(6), pp. 15-18.

Barnett, W.D. and Raja, M.K. (1995) Application of QFD to the Software Development Process, *International Journal of Quality and Reliability Management*, 12(6), pp. 24-42.

Bell, J. (2006) *Doing your Research Project*, Fourth Edition, Open University Press, New York.

Boehm, B. Abts, C. and Chulani, S. (2000) Software Development Cost Estimation Approaches – A Survey, *Annals of Software Engineering*, 10 (1-4), pp.177-205.

Carnegie Mellon Software Engineering Institute (2007, 11 January) Function Point Analysis, accessed July, 2008, from http://www.sei.cmu.edu/str/descriptions/fpa_body.html

Cobb, M. (undated) cited in Unfinished Voyages – A Follow-Up to The CHAOS Report, accessed 19th November, 2007, from http://www.standishgroup.com/sample_research/unfinished_voyages_1.php.

David, T. (2006) Dynamics of Effort Estimation (Blog), *Davids Software Development Survival Guide*, accessed 17th June, 2008, from <http://softwaresurvival.blogspot.com/2006/11/dynamics-of-effort-estimation-in-most.html>

Day, J. (2002) Software Development as Organisational Conversation: Analogy as a Systems Intervention, *Systems Research & Behavioural Science*, 17(4), pp. 349-358.

Dekkers, C.A. (2005) Creating Requirements-Based Estimates before Requirements are Complete, *Cross Talk The Journal of Defense Software Engineering*, (April), pp.13-15.

Gallivan, M.J. and Keil, M (2003) The User-Developer Communication Process: A Practical Case Study, *Information Systems Journal*, 13, pp.37-68.

Gowan Jr, J.A. and Mathieu, R.G. (2005) The Importance of Management Practices in IS Project Performance – An empirical study, *The Journal of Enterprise Information Management*, 18(2), pp.235-255.

Javed, T. Manzil-E-Maqsood. and Durrani, Q.S. (2006) Managing Geographically Distributed Clients Throughout the Project Management Life Cycle, *Project Management Journal*, 37(5), pp. 76-87.

Jiang, J.J. and Klein, G. (2001) Software Project Risks and Development Focus, *Project Management Journal*, 32(1), pp. 4-9.

Jones, C. (2004) Software Project Management Practices: Failure versus Success, *Cross Talk the Journal of Defense Software Engineering*, (October), pp.5-9.

Jones, C. (2005) Software Cost Estimating Methods for Large Projects, *Cross Talk The Journal of Defense Software Engineering*, (April), pp. 8-12.

Jones, C. (2008) Software Tracking: The Last Defense against Failure, *Cross Talk The Journal of Defense Software Engineering*, (April), accessed 17th June, 2008, from <http://www.stsc.hill.af.mil/crosstalk/2008/04/0804Jones.html>

Jorgensen, M. (2007) Estimation of Software Development Work Effort: Evidence on Expert Judgement and Formal Models, *International Journal of Forecasting*, 23(3), pp.449-462.

Jorgensen, M. and Grimstad, S. (2008) How to Avoid Impact from Irrelevant and Misleading Information when Estimating Software Development Effort, *IEEE Software*, (May/June), pp. 78-83.

Kautz, K. and Nielsen, P.A. (2004) Understanding the Implementation of Software Process Improvement Innovations in Software Organisations, *Information Systems Journal*, 14, pp. 3-22.

Keil, M. Paul, G. Mähring, M. (2007) Reporting Bad News on Software Projects: The Effects of Culturally Constituted Views of Face Saving, *Information Systems Journal*, 17, pp 59-87.

Lewis, J.P. (2001) Large Limits to Software Estimation, *ACM Software Engineering Notes*, 26(4), pp.54-59.

Mahanti, R. and Antony, J. (2005) Confluence of Six Sigma, Simulation and Software Development, *Managerial Auditing Journal*, 20(7), pp.739-762.

McConnell, S.C. (2006) The Cone of Uncertainty, *adapted from – Software Estimation: Demystifying the Black Art*, accessed 17th June, 2008, from <http://www.construx.com/Page.aspx?hid=1648>

McManus, J. (2004) A Stakeholder Perspective in Software Project Management, *Management Services*, 2004 (May), pp. 8-12.

Microsoft Encarta Online Encyclopaedia (2007) Code (Computer), accessed 10th October, 2007, from http://uk.encarta.msn.com/encyclopedia_761561581/

Morgan, M.J. (1994) Controlling Software Development Costs, *Industrial Management & Data Systems*, 94(1), pp. 13-18.

Neville, C. (2005) Introduction to Research & Research Methods, accessed 11th November, 2008 from <http://www.brad.ac.uk/acad/management/external/els/pdf/introductiontoresearch.pdf>

Nienaber, R. and Cloete, E (2003) A Software Agent Framework for the Support of Software Project Management, *Proceedings of the SAICSIT*, South Africa, ACM, pp. 16-23.

PMBOK. (2000) *A guide to the project management body of knowledge*, Project Management Institute, Pennsylvania.

Quintas, P. (1994) Programmed Innovation? Trajectories of Change in Software Development, *Information Technology & People*, 7(1), pp. 25-47.

Robinson, P. and Ringer, M. (1999) Caught Disaster: Using the Layered Systems Model as a Diagnostic Tool for Wayward Software Projects, *Work Study*, 48(6), pp.211-217.

Roetzheim, W. (2005) Estimating and Managing Project Scope for New Development, *Cross Talk The Journal of Defense Software Engineering*, (April), pp. 4-7.

Ross, M. (2003) Managing Software Size, *Proceedings of the joint ISPA/SCEA Conference 2003*, pp. 1-22.

Shepperd, M. and Kadoda, G. (2001) (abstract of) Comparing Software Prediction Techniques Using Simulation, *IEEE Transactions on Software Engineering*, 27(11), pp. 1014-102, accessed 18th June, 2008, from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=965341&isnumber=20846

Shumate, K. and Snyder, T. (1994) Software Project Reporting: Management, Measurement and Process Improvement, *Association for Computing Machinery*, not available, pp. 41-45.

Smith, D. Eastcroft, M. Mahmood, N. and Rode, H. (2006) Risk Factors Affecting Software Projects in South Africa, *South African Journal of Business Management*, pp. 55-65.

Stellman, A. and Green, J. (2005) Applied Software Project Management (Ch3), O' Reilly Media Inc., California, accessed 11th June, 2008, from <http://www.stellman-greene.com/aspm/images/ch03.pdf>

The Standish Group (1994) The CHAOS Report, accessed 19th November, 2007, from http://www.standishgroup.com/sample_research/chaos_1994_1.php

Tsoi, H.L. (1998) A Framework for Management Software Project Development, *Software Quality Institute*, Griffith University, San Antonio, pp. 592-597.

Validity & Reliability.pps (2007) accessed 16th November, 2008.

Whittaker, B. (1999) What went wrong? Unsuccessful Information Technology Projects, *Information Management & Computer Security*, 7(1), pp.23-29.

APPENDIX A

Interview Questionnaire

Interview for Bespoke Software Estimation Research in South Africa

I, Sebastian Wade, (the researcher) would like to assure you that that your responses to these questions will not be directly attributed to you or your organisation in the write up of the analysis, they may however, be combined with other comments to support the analysis.

Respondent			
Designation			
Company name			
Contact details			
E-mail address			
Date of interview			
Interview recorded	Y	N	

Company Size

SME	Medium	Large	Corporate	Multinational
-----	--------	-------	-----------	---------------

Sample Group

Respondent Number

Estimation Professional/Developer/ Implementer/Project Manager

--

General

1. (a) Which factors do you believe are leading to the high rate of software development project failure where failure is defined as being over budget and schedule?

i	Poor project management	
ii	A weak business case	
iii	Lack of top management involvement and support	
iv	Poor software estimation (cost & effort)	
v	Poor measurements	
vi	Poor milestone tracking	
vii	Poor change control	
viii	Poor Quality control	
ix	Other (Specify)	

- (b) Why these factors specifically?

i		
ii		
iii		
iv		
v		
vi		

2. (a) Do you practice software project management (SPM) in full, in part or not at all?

In full	In part	Not at all
---------	---------	------------

- (b) If in part, which parts?

Parts	
Scope and Objective Identification	
Planning	
Evaluation	

Project Development Approaches	
Software Effort and Cost Estimation	
Activity Planning	
Monitoring and Control	
Risk Management	
Resource Allocation and Control	
Managing Contracts	
HR	
Quality	
Other (Specify)	

(c) Why these parts as opposed to all?

3. Are you familiar with software estimation and its application?

Yes	No
-----	----

4. To what extent have you been or are you involved with a software development project in terms of cost to the customer (Rands)?

R1 – R500 000	R500 000 – R 1000 000	R 1 000 000 and above
---------------	-----------------------	-----------------------

5. Do you have some experience in custom/bespoke development?

Yes	No
-----	----

6. Have you been a member of a project team that is responsible for the scoping of a software development project, specifically the effort required and anticipated costs?

Yes	No
-----	----

Software Estimation

A: The Role of Estimation and Estimation Techniques

1. What do you believe the role of software estimation is within software development projects?

2. (a) In your opinion, which of the following describes the most important purpose that a software estimate makes to a software development project? (List)

		Y/N	Comment
i	Budgeting (for accuracy)		
ii	Risk analysis (cost vs. schedule tradeoffs)		
iii	Planning and Control (cost & timelines)		
iv	Software Improvement & Investment Analysis (cost benefits to these)		
v	Other		

- (b) Who do you believe benefits from the estimate?

Software provider	Software Acquirer	Other
-------------------	-------------------	-------

3. (a) Do you currently, or would you consider employing a software estimation technique? Why or why not have you chosen to do so?

Yes	
No	

- (b) If not, have you previously employed an estimation technique? Why was it abandoned/ no longer made use of?

Yes	
No	

4. (a) Which estimation technique(s) have you or would you employ? Why that technique or those techniques?

(b) Do you believe that this technique(s) is the preferred method of software estimation?

Technique		Y/N	Reasons
i	Model Based		
ii	Expert Based		
iii	Learning Orientated		
iv	Dynamics Based		
v	Regression Based		
vi	Bayesian		

5. What has been your experience in the South African marketplace of the use of estimation techniques?

6. Based on the above experience has any estimation technique appeared prevalent in the same market place?

Technique		Reasons
i		
ii		
iii		
iv		

B: Estimation Influences

1. (a) Which project factors would you consider when creating a software estimate?
(b) Which project factors would you ignore? Why?

Project Factors	Y/N	Reasons
Scope		
Technical Requirements		
Other Skills Requirements		
User Requirements		
Risk		

2. Are there any other considerations aside from those above that you would take into account when creating a software estimate?

i	
ii	
iii	
iv	
v	

3. There is evidence that estimates are often too high or too low, in your opinion what are the reasons for this?

High		Low	
i		i	
ii		ii	
iii		iii	
iv		iv	
v		v	

4. What level of involvement does the customer have in the estimate?

Fully involved	Partially involved	Not involved
----------------	--------------------	--------------

5. (a) In your experience, when compiling an estimate are you ever made aware of the clients financial expectations or constraints?

Yes	No
-----	----

- (b) Do you deem this to be an influencing factor in establishing the estimate? Why?

Yes	
No	

C: Estimate Values

1. What do you believe are the greatest difficulties in creating an accurate software estimate? Why?

Accuracy of input variables – (Software Size, complexity and technical requirements)	
Availability of input variables	
Ability to apply historical databases	
Completeness of requirements (Functional, Non functional and technical)	
Level of Uncertainty	
Risk Factors	
Other	

2. Are there any other factors that you believe assist in creating an accurate estimate?

i	
ii	
iii	
iv	
v	

3. What do you think the key component of a software estimate is? Why?

Component	Reason

4. In your opinion should software estimates be revised during the course of the software development project? When and why / why not?

Project progress	Y/N	Reason
20%		
40%		
60%		
80%		

D: Estimation Best Practices

Consider the following point of view:

“Despite a combination of estimation techniques being advocated by academics, this is seldom practised in reality.”

1. (a) Are you familiar with the concept of combining estimation techniques?

Yes	No
-----	----

- (b) Would you agree with the statement?

Yes	No
-----	----

2. Do you believe that using more than one technique would provide a more reliable estimate? Why?

Yes	
No	

3. In reality do you think it is possible to employ a combination of estimation techniques? Why?

Yes	
No	

4. Are there any “best practices” that you employ when using estimation techniques or that you are aware of? What are they? (List)

		Y/N
i	Avoid conflicting estimation goals (price/cost vs. project effort required)	
ii	Estimators must justify and criticise their estimates (remove pressures exerted by different stakeholders)	
iii	Estimate top-down and bottom-up (Customer needs and project difficulty)	
iv	Use documented data from previous development tasks (re-use of software design & components)	
v	Find estimation experts with relevant domain background and good track records (application of historical data and matching contexts)	
vi	Combine Estimates from different experts and estimation strategies (do not favour one technique understand differences between their results)	
vii	Asses the uncertainty of the estimate (changes over time, represent estimate as a probability distribution not single value)	
viii	Provide estimation training opportunities (education of the customer as to development processes for acceptance)	
ix	Other	

5. Considering the ‘Best Practices” just discussed has your perception of combining estimation techniques changed at all? Why?

Yes	
No	

6. Finally, are there any other observations or comments you would like to make regarding software estimation in South Africa that has not been covered within this questionnaire?

END

APPENDIX B

Consistency Matrix

This research aimed to establish how South African practitioners are approaching the challenge of software development estimation when dealing with bespoke development projects.					
Sub-problem	Literature Review	Hypotheses/Propositions/Research questions	Source of data	Type of data	Analysis
N/A	(Lewis 2001; Jorgensen 2004; Stellman and Greene 2005)	Proposition 1: Expert estimation techniques are preferred to model-based techniques in South Africa.	“Software Estimation” Section A: Estimation Techniques, questions 1-4	N/A	N/A
	(Stellman and Greene 2005; Jorgensen 2007; Jones 2008; Jorgensen and Grimstad 2008)	Proposition 2: Business interests as opposed to project scope are the greatest influence in arriving at software estimates.	“Software Estimation” Section B: Estimation Techniques, questions 1-3		
	(Ross 2003; McConnell 2006)	Proposition 3: Estimate values are committed to at the start of the project and are seldom revised.	“Software Estimation” Section C: Estimation Techniques, questions 1-5		
	(Boehm, Abts and Chulani 2000; Jorgensen 2004; Jones 2005)	Proposition 4: Despite a combination of estimation techniques being advocated by academics this is seldom practised in reality.	“Software Estimation” Section D: Estimation Techniques, questions 1-4		